APPENDIX C

DOCUMENTATION FOR THE SAPRC ATMOSPHERIC PHOTOCHEMICAL MECHANISM PREPARATION AND EMISSIONS PROCESSING PROGRAMS FOR IMPLEMENTATION IN AIRSHED MODELS

William P. L. Carter

Prepared for

California Air Resources Board

Contract No. A5-122-32

October 1988

Statewide Air Pollution Research Center
University of California
Riverside, CA 92521

ABSTRACT

This report documents software which has been developed for the preparation of chemical mechanisms and the processing of emissions input data for use in airshed modeling applications. This software eliminates the need to "hard wire" chemical mechanisms into airshed model programs, making it much easier to update or modify chemical mechanisms used in airshed models. It also allows the user to readily update the parameters used for lumped species in the chemical mechanism so that they best represent the distribution of organics emitted into the airshed being modeled, and provides flexibility in how emitted organics are represented in the chemical mechanism. The software includes the following:

(1) The chemical mechanism preparation program compiles a chemical mechanism into a form which can be utilized by an airshed model. Its input are text files giving the chemical mechanism in a flexible format, and its output include data files containing the mechanism-specific data, and FORTRAN source files containing mechanism-specific subroutines.

(2) Two programs were developed for processing emissions data for airshed models. The first inputs a representative emissions profile, data files giving assignments of mechanistic parameters to the emitted species, and a "lumping control file" specifying how emitted species are lumped in the mechanism, and outputs a data file giving the kinetic and mechanistic parameters for lumped species in the mechanism representing the emitted species, and a data file giving the lumped species assignments for the emissions classes. The former output file is used as input into the airshed modeling program, while the latter is input into the program which processes the detailed emissions input into the model. The second program is an example of a detailed emissions input processing program which uses this system.

(3) Interface modules to airshed modeling programs which incorporate the output of the preparation and emissions processing programs into airshed modeling programs are included as part of this software.

This report describes the format of the input and output data used or produced by these programs or interface modules, and discusses the software components involved. It also describes example calculations using this system. These use the chemical mechanisms and the detailed mechanistic assignments for emissions input which were recently developed by this author for use with this software.

A set of computer-readable data files are distributed with this report. These include: the FORTRAN source files for all of the software modules discussed here; preparation program input files implementing the chemical mechanisms used in the example calculations; data files giving emissions assignments and mechanistic parameters developed for use with this software; and input and output files for the test calculations discussed in this report.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The author wishes to acknowledge the following people for their contribution to this report: Mr. Bart Croes of the CARB modeling staff for providing guidance in the overall direction of this program, for helpful discussions, and for providing examples of software modules used in existing airshed models; Mr. Bart Croes and Paul Allen of the CARB for providing the California emissions data used in this effort; Ms. Susan Heffron of SAPRC for valuable assistance in the major effort of adapting and documenting the Preparation Program; Dr. Paulette Middleton of NCAR for providing EPA/NAPAP emissions data and for helpful discussions; Ms. Minn Poe and Joanne Lohnes of SAPRC for computer programming assistance; and Ms. Christy LaClaire, Barbara Crocker and Mae Minnich of SAPRC for word processing assistance in the preparation of this report.

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES

I. INTRODUCTION AND OVERVIEW

A. Introduction

Airshed models are important tools used by the California Air Resources Board (CARB) and other control agencies for predicting impacts of emissions of pollutants on resulting air quality. The gas-phase transformation mechanism is an important component of such models. In recent years, there has been continuous advances in our understanding of the atmospheric reactions of emitted organic pollutants, and as a result of this, the gas-phase transformation mechanisms for airshed models has undergone corresponding continual evolution. This means that for airshed models to be at the state of the art, the users of such models need the ability to change the gas-phase chemical mechanism they incorporate relatively readily, and without the need for extensive re-programming of the airshed model software, or for extensive testing for correct implementation. However, up to now most Eulerian airshed models currently in use have the chemical mechanism "hard wired" into the software, making updates in the mechanism relatively difficult.

To give the modeling staff of the CARB the ability to more readily change chemical mechanisms used in their airshed models, the CARB contracted to the Statewide Air Pollution Research Center (SAPRC) of the University of California in Riverside, to include the development, delivery, and documentation of model preparation software as part of an overall effort for developing and implementing an updated photochemical mechanism for use in airshed modeling. For the past several years, SAPRC has had in place software which can read text files giving chemical mechanisms in a highly flexible and relatively easy to input format. These data are used to prepare the data structures and mechanism-dependent subroutines required to use the mechanism in environmental chamber simulations and box-model type airshed calculations. The software includes the capability to automatically produce the code necessary to implement the steady-state approximation on many classes of species, which can result in significant savings of execution time and computer memory requirements needed to implement the mechanism. As part of this contract, this software is being adapted for use by the CARB modelers in implementing new chemical mechanisms in their airshed models. This document describes the mechanism preparation software, data files, and airshed model interface software which have been developed under this contract.

A related problem regarding use of airshed models and their gas-phase transformation mechanisms is the processing of emissions input into such models. Current emissions include many hundreds of different organic gases which undergo reactions in the atmosphere which can affect air quality. Such species, in general, react in the atmosphere at a variety of rates and types of reaction mechanisms, and the airshed model needs to take into account differences in their "reactivities" in affecting air quality. Chemical mechanisms used in airshed models cannot include explicit representations of the reactions of all of these species, and thus various "lumping" approaches where a small number of model species are used to represent the wide range of types of reactive organic emissions, need to be employed. This requires procedures and software for processing the detailed emissions data obtained from emissions inventories into formats which can be directly used as input of airshed models, i.e., describing the emissions in terms of the lumped species in the model. Ideally, such procedures and software should include not only the capability for calculating the total amounts of each of the lumped species required to represent a particular emissions profile, but also for adjusting the values of the kinetic and mechanistic parameters used in the model for the lumped species so that they correspond to the particular mixture they are representing as accurately as possible. However, existing emissions processing software are relatively inflexible, and, like the chemical mechanisms themselves, have generally had to have lumping approaches for specific types of mechanisms "hard wired" into them. They have no, or extremely limited, capability for determining the mechanistic parameters which best represents the mixes being processed. Therefore, as part of this contract to develop for the CARB a more flexible system of implementing new chemical mechanisms, we have developed software and procedures for a flexible system of processing emissions for use with airshed models. This software and the associated data files are also described in this document.

We have also developed an updated, condensed chemical mechanism and associated emissions processing data files which can be implemented with the software and the procedures described in this document. The data files implementing this mechanism and emissions processing approach are included with this software, and example calculations employing these data files are included in this document to illustrate the use of the software and procedures which were developed. However, the documentation of the development and evaluation of the chemical mechanism itself, and of the chemical assumptions and estimates which were used to derive the data files used for emissions processing, are given in a separate document (Carter 1988).

B. Overview of Components

The components described in this document consist of 135 computer-readable data files, which are listed in Appendix A. As indicated there, these consist of:

- FORTRAN source files for the model preparation program.
- FORTRAN source files for two programs which can be used for processing of emissions data.
- FORTRAN source files for three modules (subroutines) to be included in an airshed modeling program, which serve to implement mechanisms processed by the preparation program in the airshed model.
- A FORTRAN source file for a fourth module (subroutine) for an airshed modeling program which serves as an interface between the emissions processing programs and the airshed modeling program.
- Input files to the model preparation program which implement the latest version of the gas-phase atmospheric transformation mechanism developed at SAPRC (Carter 1988.), which is an update of the mechanisms previously described by Lurmann et al. (1987).
- Data files giving preliminary kinetic and mechanistic parameters and lumping assignments for the major organic species which are emitted into California airsheds.

- Output files produced by running the model preparation program on the latest version of the condensed SAPRC gas-phase atmospheric transformation mechanism.

- Input and output files, for example, of single-cell, constant light intensity calculations utilizing the software, the chemical mechanism, and the emissions assignment data files described in this document.

- A text file consisting of this document.

The current version of this software is written in FORTRAN-77, and has been developed and tested on SAPRC's PDP-11/73 computer using the RSX-11M+ operating system. The programs should also run on VAX/VMS computers with little or no modification. On the other hand, they will almost certainly need some adaptation to run on other computers, particularly for system-specific operations such as opening files, etc. However, we believe that this adaptation should be relatively straightforward for experienced programmers, and should not require a detailed understanding of the operations of this software. But since this software has not yet been implemented on the ARB computers or implemented in any full airshed model, this must be regarded as a preliminary version which is being submitted for evaluation purposes.

1. Mechanism Preparation Program

A major component of the software discussed in this document is the mechanism preparation program. As indicated above, the purpose of this program is to compile the chemical mechanism into a form which can be utilized by the airshed model. It needs to be run only once for each chemical mechanism, and re-run only if changes are made to the mechanism beyond simple modification of default parameter values. The major input to this program are files specifying the reactions, species, and default parameters in the mechanism. Its major output are files containing the mechanism-specific data required by the airshed modeling program, and FORTRAN source files for mechanism-specific subroutines required by the airshed model program to use the mechanism. The latter are subsequently compiled and linked as part of the airshed model program itself. Also included as part of this software are mechanism-independent interface subroutines to the airshed model program which can be used to implement mechanisms prepared by the preparation program. These include subroutines to read and initialize the mechanism-specific data and output a listing documenting the mechanism, subroutines to update the temperature, and light-dependent rate constants. (Also included is a subroutine to read in lumped parameter values derived from the emissions processing programs to represent overall emissions into the modeling region; this is discussed separately in conjunction with the discussion of the emission processing system.) An overview of this software and the data files they use is given in this section. The use of the mechanism preparation program and the input data it uses is documented in Section II, and the components of the software is described in more detail in Section IV.

The relationships between the programs and data files involved in the preparation of chemical mechanisms for use in airshed models, using this model preparation program, is illustrated in Figure 1. The specific components of this system are as follows:

Preparation Program Input File. The mechanism.PRP file is the main input file to the model preparation program, which gives all the data necessary to specify a chemical reaction mechanism. The name of this file consists of a one-to-eight (or fewer, depending on system file naming capabilities) character name, which is used to identify the chemical mechanism, followed by the extension ".PRP". (Other computer systems may have to use different file name conventions; the discussion in this document uses the file name conventions used on the DEC computers and used in the development of this software.) This "model name" is used to name the preparation program output files, as indicated below. The format of this input file is described in detail in Section II, and thus is not discussed further here except to note that it can include commands telling the preparation program to take input from other files, if desired. It is the current practice at SAPRC to split up model preparation input into separate .RXN files, which give portions of the mechanism, as described below, with the .PRP file consisting primarily of specifications of the overall parameters controlling the preparation of the mechanism (such as the default temperature, units for the rate constants, etc.), and then

```
                    +-----------------------------+
                    !      mechanism.PRP          !
                    !                             !
                    ! Preparation Program Input   !
                    !Defining Chemical Mechanism!
                    +-----------------------------+
                                  !
                                  V
+-----------------+      +---------------+
!  mechanism.PRO  !      !               !       +------------+
!                 !<-----!  PREPARATION  !----->! Temporary  !
!Mechanism Listing!      !    PROGRAM    !<-----! Work Files !
!     Output      !      !               !       +------------+
+-----------------+      +---------------+
                              !     !
          +-------------------+     !
          !                         !
          V                         V
+-----------------+      +------------------+
!  mechanism.MOD  !      !  mechanism.FTN   !
!                 !      !                  !
! Mechanism Data  !      !Mechanism Specific!
!      File       !      ! Subroutine Code  !
+-----------------+      +------------------+
         !                        !
         !                        V
         !                 +---------------+       +------------------+
         !                 !    FORTRAN    !       !   Airshed Model  !
         !                 !   COMPILER    !<------!Modules to Process !
         !                 !  AND LINKER   !       !Prepared Mechanisms!
         !                 +---------------+       +------------------+
         !                         !
         V                         V
+---------------------------------------------------+
!                AIRSHED MODELING PROGRAM            !
+---------------------------------------------------+
```

Figure 1. Overview of relationships between data files and programs used in the preparation of chemical mechanisms for airshed model calculations.

gives a list of .RXN files with the various components of the mechanism in a modular manner.

Chemical Mechanism Module Files. The module.RXN files are not, strictly speaking, required by the preparation program software. By convention, they consist of various modular subsets of the mechanism, and are referenced by the main input (.PRP) file, which tells the preparation program to use these files as input. The format of the input data expected for processing by the preparation program is the same regardless of whether input comes directly form the .PRP file, or from a .RXN file referenced in the .PRP file. For example, the file NOXAIR.RXN contains the inorganic reactions in the mechanism, RO2.RXN contains the general peroxy radical reactions, ALKPRODS.RXN contains the reactions of the alkane and alkene photooxidation products, etc. This modular organization assists in the mechanism development and testing process, since it permits subsets of the full mechanism to be readily assembled for testing purposes. It also simplifies updating of the mechanism, when only some of the components need to be updated. As discussed in Section II, the preparation program allows for conditional processing of input data so reactions common to different species can be included in different files, and yet be processed only once when these files are combined.

Preparation Program. The model preparation program processes the data in a model.PRP file, and uses these to prepare a model.MOD and a model.FTN file, which are required for implementation of the mechanisms in an airshed model, and a model.PRO listing file. These three types of output files are described below. It also creates and deletes several temporary work files, discussed below. The preparation program is a non-interactive program. The only direct interaction it has with the user is to ask for the name of the "model" (i.e., chemical mechanism) being processed, and to give messages saying when it is finished and whether errors occurred. The "model" name it asks for is the name used for the .PRP input file. This name is also used for the .MOD, .FTN, and .PRO output files. The operation of this program and its specific input requirements are discussed in Section II, and the various components of the model preparation program software are described in Section IV.

Temporary Files Used by the Preparation Program. The normal operation of the preparation program involves creation of several temporary files, which are normally deleted when the program terminates, but may remain if the program is aborted abnormally. These include: PREP.TMP, VCOPRP.TMP, and photname.PTM. The first two are used to save input data for subsequent processing by other parts of the program, and the .PTM files are used to temporarily save the absorption coefficient and quantum yield data for photolysis reactions. All .TMP and .PTM files can be deleted if they are found to exist after an abnormal termination of the preparation program.

Preparation Output Listing File. This file is produced by the preparation program, and has the name model.PRO, where "model" is the name used for the input file for the chemical mechanism. This ".PRO" file documents the mechanism produced by the preparation program, and consists of a listing of the mechanism, species, kinetic parameters, and other information concerning the mechanism and how it is processed, together with any warning or error messages which might result from improper input. The output in this file are in 132-column format, and the normal usage is for this file to be printed after the preparation program is run. This file is produced for the information of the user only, and is not, strictly speaking, required for the implementation of the mechanism in the airshed model. If a severe error is found in the input data during the processing of the preparation program, model preparation is aborted, and the .PRO file is the only output file produced by the program. The user can then examine this file to determine the source of the error(s) which prevented the model from being prepared. If the model preparation is successful, the .PRO file can be printed then deleted.

Chemical Mechanism Data File. This file is also produced by the preparation program, and has the name model.MOD, where again "model" is the name used for the input and output files of the chemical mechanism. This file contains the processed data for the chemical mechanism which is required (in part) for using this mechanism in airshed models, and also for documenting the mechanism in the airshed model output. These data are read by the RDMOD.FTN subprogram to the airshed model, as described below and in Section IV.C.2. This file includes such information as the default values of the kinetic parameters and stoichiometric coefficients, names of the species in the mechanism, reaction labels and descriptions, etc. It can also include default molecular weights, carbon numbers, etc., of species in the mechanism, if that information is supplied to the preparation program. The specific data in this file, and the format of these data, are described in Section IV.B.2.

Source File for Mechanism-Dependent Subroutines. The model preparation program also produces a .FTN file (named model.FTN, where "model" is the name of the mechanism files), which consists of FORTRAN source code for three model-dependent subroutines which must be included as modules in the airshed model to implement the chemical mechanism. These three subroutines are called "CONSTR", "BLDUP", and "DIFUN." CONSTR is used to initialize parameters (if required), and should be called by the airshed modeling program after all the initial concentrations and mechanistic parameter values have been read into the program, but before the simulation starts. "DIFUN" is the main model-dependent subroutine used in the model calculation, and it is called at each time step to (1) calculate the rates of all the reactions, (2) to calculate the concentrations of the steady state species, and (3) to calculate the formation rates and destruction rates of the active species. "BLDUP" is an optional subroutine which can be used, after "DIFUN" is called, to calculate the rates of formation of species which are formed in the mechanism, but which do not undergo subsequent reaction. This is optional because the calculated concentrations of these "build-up only" species do not affect any of the other model calculations, and thus it is not necessary that their concentrations be calculated if they are not of intrinsic interest. These subroutines, and considerations involved in interfacing them into an airshed modeling program, are discussed in Section IV.B.3 of this document.

Mechanism-Independent Airshed Model Subroutines. The source code for four subroutines are included to assist the ARB modeling staff in implementing chemical mechanisms processed by the preparation program in the airshed model software. These are intended to serve as models for subroutines to the airshed model program which implement the chemical mechanism. These four subroutines are in the files RDMOD.FTN, NEWRK.FTN, NEWPHK.FTN, and RDPRM.FTN, and all three use the COMMON blocks, parameters, and data specifications given in the file MODLSPC.FTN. The subroutine "RDMOD" is used to read the data in the .MOD file produced by the preparation program so these are available to the airshed model, initialize the chemical mechanism data and the rate constants, and output a listing of the chemical mechanism for documentation purposes. The subroutines "NEWRK" and "NEWPHK" are used to calculate the rate constants for temperature-dependent and photolysis reactions, respectively, and should be called every time the temperature or the solar zenith angle changes. The subroutine "RDPRM" reads the emissions-specific parameters for lumped species in the mechanism, output by the emissions processing software, as discussed in more detail in Section III and IV.C.3. These subroutines, along with the model-dependent subroutines produced by the preparation program (which are used primarily to calculate the rates of change of concentrations of active species due to chemical reaction) constitute the major chemical mechanism-dependent modules of the airshed model. These are written with no specific airshed model software in mind, but such that they can either be incorporated into any airshed model software which is sufficiently modular in design, or to serve as examples of how chemical mechanisms processed by the preparation program might be implemented as components into an airshed model. These subroutines, and considerations involved in their implementation, are described in more detail in Section IV.C.

2. Emissions Processing System

The second major component of the software discussed in this document is the system of programs and data files used for the processing of emissions data for use with lumped chemical transformation mechanisms. Processing of emissions data for use with airshed models involves a number of steps, but the system discussed here is based on the assumption that the emissions have already been processed to the extent that the emissions into the various cells of the modeling region have already been compiled in terms of mass per unit time of the various SAROAD emissions classes, based on the scheme used by the CARB and the EPA. [A discussion of the SAROAD classification scheme is beyond the scope of this document, but Carter (1988) gives a listing of the SAROAD classification scheme currently employed by the ARB and the EPA.] It is also assumed that a "typical" or "representative" emissions profile for the modeling region as a whole has been compiled, and that this "typical" profile is also given in terms of mass per unit time of emissions of the SAROAD classes. A discussion of the steps required to process the data to this extent is beyond the scope of this document. Users of existing airshed models which utilize emissions classified by the SAROAD scheme would necessarily have such software up and running in any case. Given that the emissions have already been processed at least to that extent, the system of programs and data files can be used to complete the processing required to utilize the emissions data in models using lumped mechanisms of the type developed by SAPRC and others.

An overview of this emissions processing system is presented in Section III.A of this document, and the remainder of Section III discusses the use of this system, and the data files it requires, in more detail. Briefly, the operation of this system involves two major steps. The first step involves running a program we call "PREPEMIT." This program controls the methods used in the mechanism to represent the emitted organics, and derives parameters for species in the mechanism to best represent the distribution of organic emissions into the modeling region as a whole. The input to this program includes the "typical" emissions profile representing the region as a whole, data sets giving assignments of "detailed model species" and other necessary information for the SAROAD classes (such as molecular weights, etc.), data sets giving the mechanistic and kinetic parameters for the detailed model species used in these assignments, and a "lumping control" data set describing how the model species are to be lumped. The outputs of this program include a data file giving the values of the lumped mechanistic parameters which best represent the "typical" emissions profile, and a data file giving the lumped classification assignments for the SAROAD classes. The PREPEMIT program must be run every time there is a change in the "typical" emissions profile, the SAROAD classification, the lumping approach used in the mechanism, or the assignments of kinetic or mechanistic parameters for the individual emitted organic species.

The second step in the emissions processing system involves processing the detailed (space- and time-resolved) emissions data into a form which can be directly utilized by the airshed model. Included with this software is an example program, LUMPEMIT, which demonstrates how this is done. The input to this program is the mass emissions of the SAROAD classes into a particular cell at a particular time of the airshed model simulation, and the lumped model species assignments of the SAROAD classes output by the PREPEMIT program. The output of this program is the mole emissions of the lumped species in the mechanism corresponding to the input emission profile, which can be input into an airshed modeling program.

Since the exact format of the detailed emissions data base and the processed emissions input requirements of the specific airshed model will, in general, vary, the LUMPEMIT program included with this software will almost certainly have to be modified to conform to the data format requirements utilized by the existing emissions processing and airshed model software. Alternatively, the two major subroutines of the LUMPEMIT program -- an initialization routine which reads the SAROAD assignments and the model species index data, and one which does the conversions for the specific emissions profiles) can be incorporated into existing emissions pre-processing programs utilized with the existing airshed models. The internal documentation and modular structure of the LUMPEMIT program is such that this should be relatively straightforward if the existing emissions processing programs are reasonably modular in design.

An additional component of this system is the RDPRM airshed modeling subroutine. This subroutine, which should be called during the mechanism initialization process if emissions are processed using the PREPEMIT and LUMPEMIT (or equivalent) programs, reads the lumped mechanistic parameters derived for the "typical" emissions profile, and output by the PREPEMIT program. It also reads the list of model species used in the emissions profiles in the order they are output by the LUMPEMIT program, and sets up the pointer arrays necessary to relate the processed emissions data to the species in the mechanism.

A diagram of the interrelationships between the various types of data sets and programs in this emissions processing system is given in Figure 2. A discussion of the operations of these programs and the use and formats of the various types of data which are requried in this system is given in Section III. The software components in the PREPEMIT and LUMPEMIT programs and the RDPRM interface subroutine are discussed in Sections IV.B and IV.C.3, respectively.

```
                    +--------------------------------------------------+
                    !                    Emissions Data                !
                    +--------------------------------------------------+
                           !                                   !
                           V                                   V
                    +-----------------+              +-----------------+
                    ! Typical or      !              ! Emissions Input !
                    ! Representative  !              ! For Individual  !
                    ! Profile         !              ! Cells in Model  !
                    +-----------------+              +-----------------+
+----------------+         !                                   !
! Detailed Model !         !                                   !
! Species Assign-!         !                                   !
! ments for      !         !                                   !
! SAROAD Classes !------+  !                                   !
+----------------+      !  !                                   !
                        !  !                                   !
+----------------+      V  V                                   V
!Lumping Control !   +-------------+                    +-------------+
!     File       !--->!             !  +----------------+ !             !
+----------------+   !  PREPEMIT   !  ! Lumped Species ! !  LUMPEMIT   !
                     !  PROGRAM    !-->! Assignments for!-->! PROGRAM    !
+----------------+   !             !  ! SAROAD Classes ! !             !
! Mechanistic    !--->!             !  +----------------+ !             !
! Parameters for !   +-------------+                    +-------------+
! Detailed Model !          !                                  !
! Species        !          !                                  !
+----------------+          V                                  V
                  +-----------------+             +--------------------+
                  ! Lumped Species  !             ! Emissions Input    !
                  !   Parameters    !             ! of Lumped Species  !
                  +-----------------+             +--------------------+
                           !                                  !
                           V                                  V
                 +----------------------------------------------------------+
                 ! (Initialization)                     (Emissions Input)   !
                 !                                                          !
                 !                  AIRSHED MODELING PROGRAM                !
                 !                                                          !
                 +----------------------------------------------------------+
```

Figure 2. Overview of the relationships between data files and programs used in the emissions processing system.

II. USE OF THE MODEL PREPARATION PROGRAM

A. Overview of Preparation Program Usage

The preparation program is used to prepare a chemical mechanism for use in an airshed model. It needs to be run only when a new chemical mechanism is to be implemented in the airshed model, or if changes need to be made to the mechanism beyond simply changing values of rate constants or parameters. The specific steps involved in using this program to prepare a new chemical mechanism for use in an airshed model are as follows:

(1) Using a text editor, prepare or modify the .PRP and .RXN files which define the chemical mechanism. If the chemical mechanism has been developed elsewhere, reformat the mechanism to correspond to the format expected for input data for the model preparation program, as described in Section III of this document. If the chemical mechanism input files have been obtained from SAPRC, no reformatting should be needed.

(2) Run the model preparation program. The program will output the prompt:

```
NAME FOR INPUT/OUTPUT FILES =
```

Give the model name (i.e., the name of the model.PRP file, not including the ".PRP" part of the name) in response to this prompt. No other input is required at the user terminal or (if run in batch mode) in the batch job file. The model.PRP file will then be used as input and a model.PRO and (unless serious input errors were encountered) a model.MOD and model.FTN file should be produced as a result of the operation of this program. When complete, and if no errors are encountered, the program will give the message

```
MODEL 'name           ' PREPARED.
```

will be output. If errors occurred, it will tell how many different times errors were encountered, and instead of saying the model was "prepared", it will say the model was "processed", to indicate that .MOD and .FTN files were not produced.

(3) Examine the model listing (model.PRO) files for appropriate implementation of the mechanism and for possible errors, including errors which might not have been detected as such by the program. An examination of the listing of species in the mechanism is the first step in the process, since it usually gives the most obvious indication of many types of errors. For example, if the name of a reactant was misspelled in a reaction, one will see the misspelled species name on the list of reactants. If the reactions of a reactive species has somehow been omitted, one will see it inappropriately included in the listing of the "build-up only" species if it has reactions forming it, or on the list of "dummy species" if there are also no reactions forming it. If there are no errors evident from the listing of reactants, the listing of the variable coefficient names and default values, and the reactions and kinetic parameters, could then be checked, if desired. The index numbers used for the reactants and the coefficients assigned to each reaction (with the numbers for the reactants being positive and those for the coefficients being negative) are output to the right of the listing of each reaction at the time the reactions are being processed.

(4) Compile the mechanism-dependent subroutine produced by the model preparation program (the model.FTN file), and link it with the rest of the modules of the airshed model. These include the three subroutines included with this distribution (or their equivalent), as well as the many other modules required in such models. This results in an airshed model program which incorporates the chemical mechanism which has been prepared.

(5) When the airshed model prepared in this manner is run, be sure that the model.MOD file is available to it. This file must be read by the RDMOD subroutine to the airshed model (or its equivalent) every time the program is run, since it contains mechanism specific data.

All of the information concerning the chemical mechanism and the parameters which control the defaults for the mechanism being prepared are given in the model preparation input (.PRP) file. The general format of the data in the model preparation input file is described in the following section.

B. Format of Preparation Program Input Data

The preparation program input file is a text file which consists of records which are read sequentially by the preparation program in order to obtain the parameters, commands, and input data it needs to prepare a model. One "record" is equal to one line of text and can be considered to be analogous to one computer card. Like a computer card, the maximum record length is 80 characters. In general, one command or model specification input statement is included on each record, though some types of input can be continued, as discussed in the appropriate sections. Model preparation input files can be created and modified by using any standard ASCII text editor.

Comment records can be included at any point in the input file. These are indicated by an exclamation point ("!") as the first character in the record. Comment records are ignored by the program, but are useful for including documentation concerning the model in the input data set (e.g., references for rate constants, etc.).

The input expected by the preparation program consists of three groups of records, each of which is required. These are: (1) the title record, (2) parameter records, (3) model specification records (which consist of most of the input data), and (4) the END record. The input requirements and options for the title, parameter, and end records are relatively straightforward, and are described in full in this section. The types of input and options available for the model specification input records are more complex, and these are only summarized in this section. The specific types of input and related considerations are described in more detail in the following sections. Special input options, including input inserted from other files, special input and output control commands, and conditional input for model specification records are also discussed in this section.

1. Title Record

The first noncomment record in the file is the model title, which is a brief, 64-character description of the model. If more than 64 characters are in the record, the extra characters are ignored. The contents of this record have no effect on the execution of the program, except that the title is included as part of the header in the PRO output file. Note that if a record intended to be a command or option is the first noncomment record encountered, it is assumed to be the title, and the command or option is not processed. The only exceptions to this are special commands which start "@" character, described in Sections II.B.13 and II.B.14, below.

2. Parameter Input

Parameters affecting the operation of the program and defaults of the mechanism are specified in the records immediately following the title record. Parameters can be input in any order, and may be omitted altogether in some cases. The parameter name must begin in the first column, followed by one or more blanks or tab characters, and the numerical value associated with it. Parameter input is terminated (and model specification input begun) by a record starting with a period ("."). The three parameters which are recognized by this program are discussed below. (A third option, "DEBUG", is recognized by the software, but in this version it has no effect on the operation of the program.)

Two of the parameters which are recognized by the version of the program concern temperatures used when calculating rate constants for thermal reactions. These are called TEMP and TREF. The format of the input for these is as follows:

```
TEMP temperature
```

or

```
TREF reftemp
```

where "temperature" or "reftemp" are temperatures in degrees K. The TEMP parameter sets the default temperature for the mechanism. This is passed to the airshed model program in the .MOD file, and (if the RDMOD module of the airshed program is used to initialize the mechanism) is the temperature used to calculate the initial values of the thermal rate constants, which are output with the mechanism documentation. This parameter has no effect on the predictions of the airshed model using this mechanism if temperatures which override this default are input into the airshed model. If the TEMP parameter is not specified, a default temperature of 298.12 K is used.

The TREF parameter is used in the extended Arrhenius expression which can be used to specify temperature-dependent rate constants. This expression uses three parameters, A, Ea, and B, where the rate constant at temperature T is given by

$$k = A\ (T/TREF)^{B}\ \exp\ (Ea/[0.0019872\ T])$$

The default value of TREF is 300.0 K. If the extended Arrhenius expression in the alternative form of

$$k = A\ T^{B}\ \exp\ (Ea/[0.0019872\ T])$$

is desired, set TREF to 1.0. The value of TREF is passed to the airshed modeling program via the mechanism's .PRO file, where it is needed to calculate thermal rate constants. However, it should not be changed by the airshed modeling program, since it would usually result in incorrect thermal rate constants being calculated.

The third parameter recognized by this program is called "LITTLE." The format for specifying this parameter is

LITTLE minimum-decay-rate

where "minimum-decay-rate" is a pseudo-first order rate constant. The units must be appropriate for the time units used in the mechanism, which for atmospheric applications is min^{-1}. This parameter is used to set the minimum consumption rate for steady state species, to avoid divide-by-zero errors which might occur at the beginning of a model simulation if its rates of formation and destruction are both zero. The mechanism dependent code adds this value to the total decay rate of the steady state species due to chemical reaction. Thus, though this parameter should usually be nonzero, it must also have a negligibly small value compared to the consumption rate of steady state species in the mechanism. The default value for this parameter is 1.0E-30, which is appropriate for atmospheric applications. Normally this need not be changed for atmospheric applications. The value of this parameter is passed to the airshed modeling program (via the mechanism's .MOD file) as the value of the first coefficient, which is also called "LITTLE." This value can thus be changed by the airshed modeling program, if desired.

3. Model Specification Input

The remainder of the input following the parameter specification records consist of records which specify the reactions, rate constants, species, variable coefficients, and special processing instructions which make up the model. In general, model specification input consists of groups of records beginning with a command record and ending when another command record (or null command record) is encountered. There is no set requirement concerning the order in which the various types of records are input as long as they come after the parameters discussed above. Also, any type of command can be used any number of times.

All command records start with a period (".") as their first character. Any record beginning with any other character (other than "!" for comment records, or "@" for special commands) is assumed by the program to be record-type specific, and the way it is interpreted depends on the last command record read (i.e., which command is active). Blank records are ignored except for some commands which are terminated by a blank line (e.g., .PHOT, described in Section II.B.8).

The types of commands and the types of input expected in the records that follow them (if any) are summarized in Table 1. The use of these commands, and the format for the data records expected to follow them (if any) are discussed in the following sections, followed by a discussion of redirected input and other special input commands.

4. Species Specification Input Format

Species type specification input is initiated by the species definition records ".ACT", ".BLD", ".STS", ".CON", or ".DUM" active, build-up, steady-state, constant, or dummy species, respectively. The meanings of these types of species, and the restrictions and other considerations that apply, are discussed in Section II.C. A species input definition record may be followed by any number of records that indicate the names for species of that type. Species input is terminated when the program encounters another command record (i.e., a record starting with a "."), or a null command record.

Table 1. Summary of Mechanism Specification Commands

Command	Use	Types of Records Expected to Follow
.ACT	Indicate species as active and optionally give default parameters (default initial concentration, molecular weights, and numbers of carbons and nitrogens)	Species names and default parameters
.CON	Indicate species as constant and assign default parameters	Species names and default parameters
.BLD	Give default parameters for species which do not react	Species names and default parameters
.STS	Indicate species in steady-state	Species names
.COE	Assign values to coefficients	Coefficient names and values
.RXN	Define reactions in model	Reaction input
.PHOT	Define photolysis data set	Photolysis data, terminated by blank record
.UNITS=PPM	Change kinetic parameter units from cm, molecule, second units to ppm, min. units for following reactions	None
.UNITS=OK	Do not convert units for following reactions (Cancels .UNITS=PPM)	None
.	Null record. Used for conditional input	None
.#C1	Insert instructions in model-dependent initialization subroutine (format 1)	Pseudo machine instruction codes
.#D1	Insert instructions in model-dependent derivative calculation subroutine (format 1)	Psuedo machine instruction codes

(continued)

Table 1 (continued) - 2

Command	Use	Types of Records Expected to Follow
.INS INIT	Insert instructions in model-dependent initialization subroutine (format 2)	Pseudo FORTRAN instructions
.INS DIFF	Insert instructions in model-dependent derivative calculation subroutine (format 2)	Pseudo FORTRAN instructions
.END	End of input (last record)	None

Species names must consist of 1-16 alphanumeric characters, with no embedded blanks. Species names may also not contain the characters "+", "=", "#", "!", "&", and the double quote """, since these are used to define reaction lists (as described in Section II.B.5). The characters "." and "@" should not be used as the first character in the species name, since these are used to indicate special commands to the model preparation program (and other SAPRC modeling software). In general, the use of non-alphanumeric characters other than ")", "(", "[", "]", ".", "-", "$", and "*" should be avoided to prevent possible conflicts with future or special versions of the SAPRC modeling software. It is also recommended that, if possible, the names for active species be held to eight characters or less, or at least that the first eight characters be unique, since this is required by special versions of SAPRC modeling software. However, this is not required by the present version of this model preparation program, where names can be up to 16 characters in length.

The species specification data following the species specification command record may have one of the following two formats:

(1) Species name alone, starting in the first position in the record, one name per record. For example:

```
.STS
HO.
O*1D2
```

indicates that the species named "HO." and "O*1D2" are in steady state.

(2) A species list in a restricted reaction-list type format, consisting of an "=" in the first position of the record, followed by the species names, each separated by a "+" and terminated by the end of the record. One optional blank between the species names and the "+" and "=" character is permitted. Neither continuation lines nor stoichiometric coefficients are allowed. For example, the commands

```
.STS
= HO. + O*1D2
```

have the same effect as those given illustrating the first format, above.

In addition, for all types of species except steady-state, the species specification input can be used to assign default initial concentrations, molecular weights, number of carbons, and number of nitrogens for each of the species. (If such input is not given, all these parameters are set to zero.) In this case, the input record consists of first the species name, followed by (a) the default initial concentration in PPM units; (b) the default molecular weight in grams per mole; (c) the default number of carbons, which can be noninteger (as may be appropriate for lumped species); and (d) the default number of nitrogens (which must be an integer). Each of these must be separated either by one or more blanks or tabs, an "=", or a comma. If the data at the end of the record are not required (i.e., defaults of zero are appropriate), they need not be given, and two or more commas can be used to skip over data where zero defaults are used. For example, the command

```
.CON
H2O=2.0E+4
```

specifies that the species named H2O is constant and has a default initial concentration of 20,000 ppm, the command

```
.ACT
PAN        0.0  121.05   2.0    1
```

specifies that the active species named PAN has a default initial concentration of zero, a molecular weight of 121.05, two carbons, and one nitrogen. Note that the input

```
.ACT
PAN,,121.05,2,1
```

would have the same effect.

The specifications of initial concentrations, molecular weight, and numbers of carbons and nitrogens are not necessary for the model preparation, but are used as a means to set defaults for the mechanism, which can be used by the airshed modeling program. If these defaults need not be set, species specification input is not necessary for active species which undergo one or more reactions, or build-up species which are formed but do not react, since the program assigns such species these types by default. However, species which are to be treated as constant or to be held in steady state must be specified as such, or they will be treated as active or build-up (depending on whether they react). Also, if it is desired to include in the mechanism species which do not appear in any reaction, they must be explicitly declared as "dummy" using the .DUM input. This is discussed further in Section II.C.

5. Reaction Specification Input Format (.RXN)

Reaction specification input is initiated by the record ".RXN", which signals that the input following consists of reaction specification records. Input records defining the reaction label, kinetics and mechanism of any number of reactions, can then be included (provided, of course, that the number of reactions is less than the maximum of reactions that the software can process, which is 400 in the present version). Reaction specification input is terminated when the program encounters another command record (i.e., a record starting with a "."), or a null command record. A minimum of one record is required for the specification of each reaction, but specifications for a reaction can be continued onto more than one record, if necessary.

Specification data for a reaction must have the following format, where the characters ")", ";", and (if comments are included) "!" are used to delimit the four different fields involved.

label) kinetic parameters ;reaction list !comments

For all the reaction definition records, the characters right parenthesis, [")"], semi-colon (";"), and (if comments are included) "!" are required by the syntax, but there is no special requirement as to their exact position in the record. Note, however, that the right parenthesis must be followed by a space, or an error will result.

The characters between the start of the record and the ")" character consist of the reaction label. The reaction label can consist of 1 to 4 alphanumeric characters, with no restriction on the characters except that they cannot contain embedded blanks (leading blanks are acceptable) and obviously the ")" character cannot be used. They should be unique for each reaction, but the program does not check for this, and model preparation proceeds without error even if duplicate labels exist. However, depending on the details of the airshed model software, it may not be possible to change rate constants or kinetic parameters for reactions which do not have unique labels without re-preparing the mechanism. If more than four characters are used for the reaction label, the program will use the LAST four characters as the label.

The data between the ")" character, which terminates the reaction label specification, and the ";" character consist of the specification of the kinetic parameters, which are used to calculate the rate constant. There must be a blank character (or a tab) between the ")" and the first character used to define the kinetic parameters. The rate constants can, in general, be dependent on temperature or on light characteristics (but not both), and there are several ways in which these parameters, and their dependences can be specified. The different formats for the specification of the kinetic parameters, and how the parameters for the various types of kinetics are used to calculate the rate constants, are discussed below. Note that, except for reactions with "FALLOFF" kinetics (see below), all the kinetic parameter definition data must be on the first record of the reaction definition input for each reaction.

The data in the field following the ";" and until the end of the record (unless continued as discussed below), or until a "!" character which indicates that comments follow, consist of the "reaction list." This gives the reactants, products, and (where applicable) stoichiometric coefficients for the reaction being defined. The format required for reaction lists are described below.

The end of a record or the "!" character normally terminates a reaction list. However, if the reaction list is too long to fit on one record, it can be continued on the following record by ending the first record with an "&" character and continuing the list on the following record. The "&" should only appear in the reaction list where a blank would be legal on the reaction list (see below). Leading blanks on a continuation record are ignored. Note that the reaction description record is limited to 80 characters per record.

A comment can optionally be placed on the reaction description line after the products, using an exclamation point "!". Anything after the "!" to the end of the record is ignored in the processing by the preparation program, though they are stored with the reaction list in the mechanism's .MOD file as part of the mechanism documentation. For example:

```
12) 1.8E-12, 2.72        ; O3 + NO = NO2&
                                   + O2  !reaction on 2 lines
```

contains both a continuation statement and a comment. Comments themselves cannot be continued, though if additional comments are desired for documentation purposes, comment records (indicated by the "!" in column 1) can be included at any point in the input file.

a. <u>Specification of Reaction Kinetics</u>

Several different options are available with regard to the specification of reaction kinetics, depending on whether the rate constant for the reaction is temperature dependent and the nature of the temperature dependence, whether it is a photolysis reaction, whether it is to always have the same rate constant as another reaction, or whether the rate constant is independent of temperature and light intensity. The format of the data in the kinetic parameter field of the reaction specification record determines which of these options are employed. These options, and the ways in which the kinetic parameters input are used to calculate the rate constants for the reactions, are discussed in this section.

<u>Thermal Reactions with Extended Arrhenius Kinetics</u>. If the reaction specification record is of one of the following formats

```
label)  A, Ea, B   ;reaction list

label)  A, Ea      ;reaction list

label)  A, , B     ;reaction list

label)  k          ;reaction list
```

where A, Ea, B, and k are decodable as real numbers, then the temperature dependent rate constant (k) of the reaction is given by the following expression.

```
k = A * (TEMP/TREF)**B * exp (-Ea/0.0019872*TEMP)
```

where A, Ea, and B are the parameters given in the kinetics field of the reaction specification record, TEMP is the temperature in degrees K, and TREF is the "reference temperature", which is by default 300 K unless it is changed in the preparation program parameter input, as described above in Section III.B.2. Note that A is the Arrhenius pre-exponential factor, Ea is the activation energy in kcal/mole, and B is a unitless quantity.

The kinetic parameters in the kinetic parameter specification field can be either separated by commas (one or more blanks) or a combination of commas and blanks (as shown in the examples). If only two numbers are given in the kinetic parameter input field, then they are assumed to specify A and Ea, with B=0, unless there are two commas separating the two numbers, in which case they are assumed to be A and B, with Ea=0. If only one number is in the field, it is assumed to be just the rate constant, and that Ea and B are zero. In any case, all three of these parameters are passed to the airshed modeling program through the mechanisms .MOD file for use by that program in calculating the rate constant as a function of temperature.

For atmospheric modeling applications, the kinetic parameters are assumed to be in ppm, minute units unless the ".UNITS=PPM" command is in effect. If a ".UNITS=PPM" command has been encountered in the input stream prior to processing this reaction, and it has not been canceled by a subsequent ".UNITS=OK" command, then the kinetics parameters in the reaction specification input are assumed to be in cm, molecule, sec units, and are converted into ppm, minute units prior to saving the kinetic parameters for use in the airshed modeling program. This is discussed in Section II.B.7, below. Note that regardless of whether the ".UNITS=PPM" option is in effect, the units of Ea are always kcal/mole.

Thermal Reactions With FALLOFF Kinetics. For thermal reactions which are not at their high pressure limit, the Troe falloff expression might be more appropriate to determine the temperature dependence of the rate constant. For such reactions, the following input format is used:

```
label) FALLOFF   ;reaction list
     A0, E0, B0
     A1, E1, B1
     F,  N
```

where A0, E0, B0, A1, E1, B1, F, and N are decodable as real numbers. These numbers must be input as shown on the three records following the input of the reaction list. If the reaction list is continued, the continuation records go BEFORE the input of A0, etc., that is,

```
label) FALLOFF   ;start of reaction list &
                  more reaction list &
                  last of reaction list
     A0, E0, B0
     AI, EI, BI
     F,  N
```

For reactions with this type of kinetics, the rate constant is obtained from the parameters using the following expression:

```
k = [k0*M/(1+(k0*M/kI))] * {F ** 1/[1 + (log10(k0*M/kI)/ N)**2]}
```

where

```
k0 = A0 * [(TEMP/TREF)**B0] * exp (-E0/0.0019872*TEMP)
kI = AI * [(TEMP/TREF)**BI] * exp (-EI/0.0019872*TEMP)
```

and where TEMP and TREF are the temperature and reference temperature as described above, and M is the total pressure, given in concentration units consistent with the units used for concentration of the reactants (ppm for atmospheric modeling applications). Note that k0 is taken as the low pressure rate constant, and kI is the high (infinite) pressure rate constant. For airshed model applications, a value of M = 1.0E+6 ppm should be used, though that is not set by the preparation program. (A value of M fixed at 1.0E+6 ppm is set in the code for the NEWPHK airshed model subroutine for calculating temperature dependent rate constants. If simulations are to be carried out at other pressures, the capability to vary M must be implemented as part of the code for the airshed model program.)

The kinetic parameters (A, E, B, etc.) can be separated by either a comma or blank(s). If B0, BI, E0, or EI are zero, they can be omitted in a manner exactly analogous to the input format for kinetic parameters for reactions with extended Arrhenius kinetics, as described above. The values of F and N must be supplied, and both must be nonzero.

Photolysis Reactions. The rate constants for photolysis reactions are calculated using the intensity and spectral distribution of the light source and the absorption coefficients and quantum yields of the photolysis reactions. The intensity and spectral distribution of the light source (i.e., natural sunlight for airshed model calculations) is determined by the airshed model software or model calculation program, and is not part of the mechanism specification data processed by the preparation program (see Section IV.C.5 for a discussion of an airshed model subroutine which can be used to derive such data and calculate photolysis rates). On the other hand, the absorption coefficients and quantum yields of the photolysis reactions is regarded as part of the mechanism, and it is the responsibility of the preparation program to supply these data to the airshed modeling program. These absorption coefficient and quantum yield data are regarded by the preparation program as "photolysis sets", which can either reside in separate files which can be read and processed by the preparation programs, or (more commonly) be included in the input stream as records following a ".PHOT" command, as discussed in Section II.B.8. Each photolysis set is identified by a 1-8 alphanumeric character label, which must be unique for each photolysis reaction which has different absorption coefficients and quantum yields.

Photolysis reactions are identified as such as having the label for the photolysis set being given in the kinetic parameter field of the reaction specification input. This can have one of two formats. If the photolysis data is to be read from a separate photolysis file, the following format is used:

```
label)  PF=photname   ;reaction list
```

where "PF=" (with no spaces) is required by the syntax, and "photname" is a photolysis set name. If this format is used, a file named "photname.PHF" must exist and contain the absorption coefficient and quantum yield data for this reaction. The format of the data in this file must be exactly the same as the format of the records following the ".PHOT" command, discussed below in Section II.B.8. If such a file does not exist, or has data in an improper format, an error will result. Note that if this format is used, the character string used for "photname", in addition to being eight characters or less, must conform to any other file name rules imposed by the operating system in use.

If the absorption coefficient and quantum yield data are to be included in the standard preparation program input stream (i.e., the .PRP file or one of the files "included" by it as discussed in Section II.B.13, below), then the second format should be used. This is,

```
label)  PHOT=photname   ;reaction list
```

where "PHOT=" (with no following blanks) is required by the syntax, and "photname" is a 1-8 alphanumeric label of length 1-8 which specifies the photolysis set. In this case, the photolysis set is not read from a separate file, but from the records following a ".PHOT photname" command which gives a "photname" which matches that in the reaction. It does not matter whether the ".PHOT" input for a photolysis set comes before or after the reaction specification input for reactions which use it. However, if no such data are input for each "photname" referenced in a photolysis reaction using the "PHOT=" format before the .END record is processed, then "file not found" errors will occur, and the mechanism preparation will fail.

More than one reaction can reference the same photolysis set (i.e., have the same absorption coefficients and quantum yields). However, if the same name is used in two reactions, one using the "PF=" form and the other using the "PHOT=" form, the program will use the SAME data for both (either from the photolysis file or from the preparation input file), depending on which form of reaction definition record came last. This is obviously not proper input, but the program will not detect this as an error. It is not recommended that use of the "PF=" and "PHOT=" forms be mixed within the same mechanism, and use of the "PHOT=" form is the preferred usage for mechanisms developed at SAPRC, since the absorption coefficient and quantum yield data can be kept in the same data files which give the specifications for the photolysis and other reactions of the photolyzing species. It may be desirable to remove the "PF=" option when implementing this model preparation software on other computers.

<u>Reactions with Constant Rate Constants</u>. If a reaction has a rate constant which depends neither on temperature or light characteristics, then it can be specified as a reaction with CONSTant kinetics. For calculations where the temperature is changing frequently, this results in slightly less processing than using the extended Arrhenius form with Ea and B = 0. The format for reactions with constant kinetics is as follows:

```
label)  CONST k      ;reaction list
```

where "k" is decodable as a real number and is the rate constant, which must be separated from the code word "CONST" by at least one blank or tab character.

Note that if the .UNITS=PPM option is in effect, then the units of the rate constant for constant reactions are converted in exactly the same way as the Arrhenius pre-exponential (A) factor for reactions with extended Arrhenius kinetics. However, for non-first order reactions, the results are different from reactions with extended Arrhenius kinetics with Ea = B = 0, in that in the latter case, the value of B changes depending on the order of the reaction (see Section II.B.7, below), and thus a slight temperature dependence for the rate constant is introduced, while the rate constants for constant reactions remain temperature-independent.

Reactions Forced to have the Same Kinetics as Other Reactions. In many cases, a series of similar reactions are assumed to have the same rate constant. If this is desired, this can be forced to be the case by using the "SAMEK" option in the kinetic parameter input field of a reaction. In this case, the input format is

```
label)  SAMEK rxlabel ;reaction list
```

where "rxlabel" is the reaction label for some other reaction, which has been previously defined. It must be separated from code "SAMEK" by at least one blank or tab character. If the reaction label referenced does not correspond to the label from some previously defined reaction, then an error message will be given and the preparation will fail. The reaction referenced must not itself have the SAMEK option in its kinetic parameter definition field. Otherwise, it does not matter whether the referenced reaction is thermal, photolytic, or has a constant rate constant.

Note that if the SAMEK option is used, no separate storage location is allocated for the rate constant of this reaction. Instead, the mechanism dependent code used to calculate the rates of this reaction use the storage location allocated to the rate constant for the reaction whose label was given following the SAMEK code. This means that if the rate constant or kinetic parameters for a reaction is changed during the execution of the airshed modeling program, the rate constants used for all reactions which reference it, using the SAMEK option, will also be changed.

b. <u>Format of Reaction Lists. Specification of Reactants, Products and Coefficients</u>

The reaction list consists of the part of the reaction specification input which gives the names of the reactants, products, and (where applicable) product stoichiometric coefficients and rate factor coefficients. The reaction list consists of up to three reactant species or rate factor coefficient names, each separated by a "+", followed by an "=", which in turn is followed by the list of product species and (optionally) product stoichiometric coefficients. There is no limit to length of the product list, except that the total number of products and product coefficients in the mechanism as a whole must be no greater than a maximum allocated in the software (which is set at six times the maximum number of reactions). Each product species name must be separated from the others by a "+". Blank spaces in the reaction list will be ignored, but embedded blanks in a species, or coefficient name are not legal. If two reaction names are not separated by a "+" or an "=" then the program will issue an error message. In addition, blank reactants, products, or coefficients (see below) are illegal. For example

```
A +  + B = C
```

is not allowed and a syntax error will result.

A reaction with no products is legal, but a reaction with no reactants is not. If zero order reactions are to be included in the mechanism, either a constant species or a reaction rate coefficient (see below) must be specified in the reactant field.

Each product species can have associated with it a stoichiometric factor, called a "coefficient." This is not the case for reactants, which must be entered separately and the appropriate number of times in the reaction list, for example:

```
7) 3.3E-39, -1.05     ; NO + NO + O2 = NO2 + NO2
```

The stoichiometric factor is indicated by the species name being preceded by the character "#", followed by a coefficient name of 1-8 alphanumeric characters, and then a single blank space which separates the coefficient name from the species name. For example, the previous example can be rewritten using a stoichiometric factor as follows:

```
7) 3.3E-39, -1.05     ; NO + NO + O2 = #2 NO2
```

where "2" is the coefficient. In this case, the coefficient is referred to as a "constant" coefficient, since it is decodable as a number, which is its value and which cannot be changed.

It is also legal to have a "variable" coefficient, which has a name which cannot be decoded into a numeric value, and whose value can be assigned using input records following the ".COE" command. The use of the ".COE" command, and the distinction between "constant" and "variable" coefficients is discussed further in Section II.B.6, below.

A stoichiometric factor can be associated with more than one product if the coefficient name (indicated by a "#" immediately preceding it) is followed by, instead of a single product name, a list of product names, each separated by a "+", enclosed by double-quote (") characters. For example, the following two reactions

```
A + B = #2 "D+E+F"
A + B = #2 D + #2 E + #2 F
```

are equivalent and both have three products and three coefficients. In addition, multiple coefficients for products are allowed, for example

```
A = #C1 B + #C1 #C2 C
```

or

```
A = #C1 "B + #C2 C"
```

are both legal and represent the same reaction. If there are multiple coefficients, they are multiplied together to obtain the stoichiometric coefficient for the product. Product coefficients must have corresponding product species, i.e., the syntax

```
A = B + #C
```

or

A = #C + B

is not legal.

Coefficients can also be included in the reactant field of the reaction list (i.e., before the "="), but the meaning and syntax of this usage is different than when they occur in the product list. As with the product list, a coefficient name is indicated by the use of a "#" character which precedes it, but unlike the product list, they are not associated with any reactant. For example, the syntax

A + #C B = PRODS

is not legal, whereas the examples

A + #C = PRODS

#C + A = PRODS

#C = PRODS

are legal. When used in the reactant field, the values of the coefficient is used as a factor which is multiplied by the rate constant when calculating the reaction rate. For example, in the reaction

C57) PHOT=KETONE ;MEK + HV + #QY.MEK = products

the rate constant for the reaction is calculated by multiplying the value of the variable coefficient "QY.MEK" by the rate constant calculated using the absorption coefficient and quantum yield data in the photolysis set named "KETONE."

Multiple coefficients for reactions, i.e., the syntax

R1) 1.0 ;A + #C1 + #C2 = PRODS

is legal. In this case, the rate constant determined by the kinetic parameters is multiplied by the values of both the coefficients to obtain the rate constant used for the reaction.

A special example of the reactant coefficient syntax is then the symbol "#RCONlabel" (where "label" is the reaction label of some previously input reaction) appears on the reactant list. In this case, the program interprets the special coefficient named "RCONlabel" as having a value equal to the rate constant of the reaction referenced by "label", and thus the rate constant of the referenced reaction is multiplied by the value determined by the kinetic parameters in order to obtain the rate constant used. ("RCONlabel" is not a true coefficient, since it is not included on the list of coefficients passed to the airshed modeling program in the mechanisms .MOD file, nor can it be used as a product stoichiometric coefficient.) This is intended to be used when reaction reference by "label" is the reverse of the other reaction, and thus the kinetic parameters specified for the current reaction actually refer to an equilibrium constant. An example of this is the following set of reactions:

```
8)  FALLOFF                    ;NO2 + NO3 = N2O5
    2.0E-30, ,-4.3
    1.4E-12, ,-0.5
    0.6     0.972
9)  1.06E+27, 22.55, -1.0      ;N2O5 + #RCON8 = NO2 + NO3
```

where the rate constant for reaction 9 is given by

```
k9 = k8 * 1.06E+27 * [(TEMP/TREF)**-1] * exp (22.55/0.0019872*TEMP)
```

where k8 is calculated by the "FALLOFF" expression discussed above.

Note that in most respects, use of reactant coefficients is equivalent to use of constant species as reactants. In both cases, the rate constant is multiplied by the values of the coefficients or constant species to obtain the rates of the reactions. Other than the fact that the values of constant species and the values of coefficients are stored in different arrays, the only practical difference is when the .UNITS=PPM option, which converts the kinetic parameter units (see Section II.B.7, below), is in effect. The way the rate constant is converted is determined by the "order" of the reaction, which is determined by counting the number of reactant species. Constant species are counted as reactants when calculating the order of the reaction, but reactant coefficients are not.

Legal reactant names consist of 1-16 characters not containing embedded blanks. Reactant names also cannot contain the characters ",", "+", "=", "#", "!", "&", or a double quote ("). Alphanumeric characters (A-Z, 0-9) and the special characters "(", ")", "[", "]", ".", "$", "-", "*", and "/" are legal; other special characters are also legal, but should be avoided to allow for future special use if the program is modified. The user should also avoid using lower-case letters and nonprinting (control) characters to avoid ambiguity as to what the name actually is when it is printed.

Legal coefficient names consist of 1-8 characters (not counting the leading # which is used to indicate that a symbol is a coefficient when it appears in the reaction list), must not contain embedded blanks, and have the same restriction of characters that applies to species names. If the coefficient name is a number, the coefficient is assigned a value of that number; otherwise they are given the default value of 0. This is discussed further in Section II.B.6.

c. <u>Examples of Reaction Specification Input</u>

Several illustrative examples of legal and illegal ".RXN" inputs are given in this section. The following examples are legal:

```
.RXN
1)  PF=NO2A                   ;NO2 + HV = O + NO     !This is k1
2)  6.0E-34, ,-2.3            ;O+O2+M=O3+M
7)  FALLOFF                   ;NO2 + NO3 = N2O5
    2.0E-30, ,-4.3
    1.4E-12, ,-0.5
    0.6      0.972
26) 8.0E-12, -0.50            ;NO + NO3 = #2 NO2
ABC)                          ;NO + NO2 + H2O&
                                        = #2 HONO
EFG) PHOT=NO2B                ;NO2 + HV = NO + O3
```

Several examples of illegal ".RXN" input are indicated below, along with indications as to why they are illegal:

```
PF=NO2        ;NO2 + HV = NO + O
    (No reaction label)

   1)  PF=NO2        NO2 + HV = NO + O
       (No ";" indicating start of reaction list)

   1)  PF=NO2->O(3P) ;NO2 + HV = NO + O
       (Photolysis file name containing nonalphanumeric characters
        and is more than eight characters long)

  11)  3.3E-39,-1.05; #2 NO + O2 = #2 NO2
       (Coefficients not allowed for reactants as stoichiometric
        factors)

  25)  1.3E 14,20.7 ;HO2NO2 = HO2. + NO2
       (The A factor contains an embedded blank)

9001)  0.0          ; NO + NO2 + H2O + WALL = #2 HONO
       (More than three reactants not allowed)

9999)  1.0E-23, , -2.34  ;REACTANT = PROD & DUCT
       (Illegal use of "&" in middle of a species name)
```

6. Coefficient Specification Input (.COE)

As discussed above, reactions can use what we designate "coefficients" to define product yield stoichiometries, or in the calculation of reaction rates. Coefficients can be "constant" or "variable". Constant coefficients consist of simple numeric values and do not have names, and variable coefficients have both names and values. Constant coefficients are defined in reaction lists (or inserted instructions, see Section II.B.9), when coefficients which can be decoded into numeric values are encountered. Variable coefficients are defined in reaction lists or inserted instructions when a character string is indicated as a coefficient which cannot be decoded into a numeric value. Since no numeric value can be associated with such a coefficient at the time it is encountered in a reaction list or inserted instruction, the ".COE" command can be used to assign a value to a variable coefficient. This command can also be used to define and give values to variable coefficients prior to their use in reaction lists or inserted instructions. If no ".COE" command is used to define the value of a variable coefficient encountered in a reaction list or inserted instruction, it is given a value of zero.

The format of the records following the .COE command must be a variable coefficient name, starting in the first column, followed by a "=", ",", or one or more blanks or tab characters, and then a numeric value. If only the name is given, it is given a value of zero. Variable coefficient names can have one to eight characters, and must not be decodable as a number. (For example, "2.0" is not a legal variable coefficient name, but "TWO" is.) Other than that, the rules for variable coefficient names are the same as for species names (i.e., they may not have characters which have special meanings when encountered in reaction lists).

For example, if the reaction description record was as follows:

```
5)  8.0E-12, -0.50        ;NO + NO3 = #VAR NO2
```

the variable coefficient "VAR" will be assigned an initial value of zero. To give "VAR" another initial value, the ".COE" command is used as follow:

```
.COE
VAR = 2.0
```

Note that this has the same effect as defining the reaction as

```
5)  8.0E-12, -0.50        ;NO + NO3 = #2 NO2
```

except that in this case a constant coefficient is used, which is automatically given a value of 2.0 (and no name), and which cannot be changed in subsequent .COE commands. The use of variable coefficients allows for changes in their values after the model is prepared, where the name can be used to reference the coefficient whose value is being changed.

7. Kinetic Parameter Units Control (.UNITS)

For atmospheric applications, the kinetic parameters should be in ppm, minute units when they are passed to the airshed modeling program. However, if it is desired to give the kinetic parameters in the reaction specification input in cm, molecule, sec units, the command

```
.UNITS=PPM
```

can be given prior to the processing of any reaction input for reactions where those units are used. This will result in the program assuming that kinetic parameters for subsequent reactions input are in cm, molecule, sec units, and thus converting them to ppm, minute units for use in the airshed model.

The effect of the ".UNITS=PPM" command is canceled by the

```
.UNITS=OK
```

command. No unit conversions will be done for any reaction subsequently input, though the unit conversions for reactions previously input, if any, will be unaffected. Note that the default is to have no unit conversions, so this command only has an effect if ".UNITS=PPM" has been given previously. There is no limit to the number of times these commands can occur in the input stream.

If the ".UNITS=PPM" option is in effect, the kinetic parameters for any thermal reaction with extended Arrhenius kinetics are converted as follows:

```
A (ppm-min units) =
  A (cm-molec-sec units) * 60 *[(7.3395E+15)/tref)] ** order
```

and

```
B (ppm-min units) = B (cm-molec-sec units) - order
```

where "order" is the number of noncoefficient reactants in the reaction. A pressure of 1 atm is assumed for the conversions for the purpose of defining the concentration of 1 ppm. Note that the activation energy is unaffected by this conversion; it is always in kcal/mole.

The conversion for a thermal reaction with FALLOFF kinetics is analogous, except that the order used for the "low pressure" parameters (A0 and B0) is one greater than the order of the reaction as it is decoded. The conversion of the rate constant for CONSTant reactions is done using the same formula as shown for A, above. No conversion is done for photolysis reactions; the absorption coefficients must always be in cm^2, base e units.

If a reaction list contains the symbol "#RCONxx" (see discussion of reaction lists, above), then the kinetic parameters for the reaction actually refer to an equilibrium constant, and thus a somewhat different conversion procedure is required. In this case, the parameters are converted by:

A (ppm-min units) =
  A (cm-molec-sec units) *[(7.3395E+15)/TREF)] ** delta order

and

B (ppm-min units) = B (cm-molec-sec units) - delta order

where "delta order" is the order of the forward reaction - the order of the reverse reaction. (The "forward" reaction is the reaction containing the "#RCONxx" symbol. The "reverse" reaction is the reaction referenced by it.)

8. Photolysis Data Input (.PHOT)

As discussed above in Section II.B.5.a, the processing of photolysis reactions requires the absorption coefficient and quantum yield data for such reactions to be input. These data can either be contained in separate .PHF files, referenced by the "PF=phffilename" kinetic parameter option, or they can be included in the main preparation program input stream, following a ".PHOT" command. In the latter case, the ".PHOT" command syntax is

.PHOT photsetname

where "photsetname" is the photolysis set name which matches a photolysis set name referenced by a "PHOT=photsetname" kinetic parameter option in some reaction either previously or subsequently input. In either case, the syntax for the input data in the .PHF file or in the records following the ".PHOT" command are the same, except that in the former case the input is terminated by either a blank input record or an end-of-file, while in the latter case the input must be terminated by a blank input record. In addition, "@filename", "@#SHOW", and "@#NSHOW" commands (discussed in Sections II.B.13 and II.B.14, below) cannot occur in .PHF files, though they can occur in the input following a .PHOT command. On the other hand, comments (records starting with an "!") are legal in both cases.

The first record in the .PHF data sets or following a .PHOT command is a title record which is ignored by the preparation program, other than copying it onto the output listing (.PRO) file.

The next record optionally contains a factor used to multiply the product of the adsorption coefficient and quantum yield. The name FACTOR or just FA must begin in the first column of the second record followed by a space and the number to be used as a factor. The factor must be greater than zero. If this record is absent, the factor is assumed to be 1.0. FACTOR records can also occur later in the input stream, if it is desired to change the factor. The new factor is then used to correct the data subsequently read in, but not the data previously processed, where the old factor was used.

The records which follow contain the wavelength (in microns), the absorption coefficient (in cm^2 $molec^{-1}$, base e), and quantum yields for each wavelength each separated by a blank space. An example of the format of a photolysis data set is as follows:

```
.PHOT NO2
NO2 PHOT [REC. BY NASA (1985), JPL PUB. 85-37] FOR T=298K.
.FAC 1.0E-20
0.275  4.020 1.000
0.280  5.540 1.000
0.285  6.990 1.000
0.290  8.180 1.000
0.295  9.670 0.984
```

.FAC 1.0E-19
0.300   1.170 0.980

etc.

If no quantum yield is given, it is assumed to be 1.0. (Alternatively, if only two numbers are given per record, the second can be thought of as the absorption coefficient x quantum yield product.) The input is terminated when a blank record (or, for .PHF files) and end-of-file is encountered. If the blank record input is following a .PHOT record, then the .PHOT input terminates and the program ignores all subsequent input until another command record (with a "." in the first column) is read in.

There is no requirement concerning the wavelength intervals that can be read in, except that all nonzero absorption coefficients and quantum yields for wavelengths above 280 nm must be specified, and the input data must be given in order of ascending wavelengths. It is legal to give wavelengths below 280 nm, though they would be ignored. If the absorption coefficient x quantum yield product is zero at 280 nm or greater, an entry with a zero absorption coefficient or quantum yield must be given for the wavelength just before the values become nonzero. A zero absorption coefficient or quantum yield entry must also be given for the first wavelength for which it is zero at the high wavelength end of the spectrum as well.

The program calculates the areas under the absorption coefficient x quantum yield spectrum for each of the standard intervals used for calculating photolysis rates, and these are the data which are passed to the airshed modeling program (via the mechanism's .MOD file). This is calculated as if the spectrum consisted of straight lines between each of the points which were input. Therefore, it is recommended that a sufficient number of points be included in the photolysis set (such a spectrum consisting of straight line segments is a good approximation of the true spectrum), and let the program calculate the appropriate areas for the various intervals. This means that photolysis files for species which have a lot of structure in their absorption spectrum should have data for more closely spaced wavelengths than those for species with smooth spectra. (The maximum number of wavelengths in photolysis files which can be processed by the program is determined by the MAXAQY parameter (see Table 11, Section IV.A.1, below).

9. Inserted Instruction Input (.#xx, .INS)

Inserted instructions are used to explicitly include special instructions into the model specific subroutines. This is useful in cases where it is desired to have values of coefficients or rate constants calculated from values for other coefficients, rate constants, or species concentrations. Inserted instructions can be included in the initialization (CONSTR) subroutine, which is called by the airshed modeling program after the parameters are read in, but before the simulation begins, or at the beginning of the mechanism specific (DIFUN) subroutine which is called at each time step to calculate the reaction rates and rates of change of the active species due to chemical reaction. Instructions should be inserted into the initialization routine, via the ".#C1" or the ".INS INIT" command, if the only operands are coefficients, rate constants, or constant species concentrations which are constant throughout the simulation. On the other hand, if the operands change during the simulation, as would be the case if an active species concentration is used to calculate a coefficient value, etc., then the instructions should be included in the DIFUN subroutine via the ".#D1" or the ".INS DIFF" instructions. The formats of the inserted instruction input are the same regardless of whether the instructions are to be inserted into CONSTR or DIFUN.

There are two formats for inserted instructions: those following the ".#C1" or the ".#D1" commands (format 1), and those following the ".INS INIT" or the ".INS DIFF" commands (format 2). The two formats have the same overall capabilities in terms of the types of instructions which can be specified. Only format 1 is compatible with versions of the preparation program which produce the mechanism specific instructions in machine code (most commonly used in mechanism development research at SAPRC). Format 2 is easier to understand, and produces more efficient FORTRAN code. The two formats can be mixed within the same input file (though this is not recommended). After a command specifying inserted instruction input of a specific format, inserted instruction input in that format is expected for the following records until another command record (i.e., a record starting with a "."), or a null command record, is encountered. The inserted instruction commands are summarized as follows:

Format 1	Format 2	Placement of Code
.#C1	.INS INIT	In the initialization subroutine (CONSTR). Executed only once.
.#D1	.INS DIFF	At beginning of code to calculate derivatives for the integration (DIFUN). Executed once for each time step of the integration solver.

Inserted instructions can reference coefficients, reactions, or species. If the inserted instructions reference coefficients which have not been defined previously, the new coefficients are added to the end of the list of constant coefficients, if they are decodable as a number, or to the end of the list of variable coefficients if not. On the other hand, if a species name is referenced, it must be referenced in a species specification statement or in a reaction elsewhere in the input. Likewise, reactions which are referenced must also be defined. However, the reaction or species does not have to be referenced or defined in the input before the inserted instruction is encountered, because the preparation program does not process inserted instructions until after all other input is processed.

a. <u>Inserted Instructions in Pseudo-Machine Format (.#x)</u>

When the inserted instruction codes ".#C1" or ".#D1" are used, the inserted instruction codes are expected to be given in format 1. In this format, the inserted instruction input is a list of pseudo-machine instructions separated by a space or ignored character (usually a comma). These can be thought as operations involving machine registers and array elements, where the latter can be coefficients, concentrations, or rate constants. The format of these pseudo-machine instructions can be one of the following:

<instruction code><1st register no.>

or

<instruction code><1st register no.><second register no.>

or

<instruction code><1st register no.><array type>'name'

where "<...>" refers to a single character of a certain type, as discussed below, and where "'name'" is the name of a species, coefficient, or reaction label, depending on the array type enclosed in single quotes. There are no spaces or other separator characters between the elements of the machine instructions. (Several examples are shown below.) There can be more than one instruction per record; if so, they must be separated by exactly one character, which is ignored (but should be a space or comma for readability). If the second character following a valid instruction is a blank, the remainder of the record is ignored.

The first character of an instruction is always the instruction code. The second character is always a number ("0", "1", or "2") which can be thought of as referring to a machine register (R0, R1, or R2, respectively) which is the first operand of the instruction. The third character or group of characters defines the second operand of the instruction, and can be either a number ("0", "1", or "2"), if the second operand is to be thought of as a machine register, or to an array type code character, followed by an array element name enclosed in single quotes. Table 2 summarizes the legal instruction codes for this format of inserted instructions, the characters which are used to designate register operands, and the legal array type codes which can be used as the second operand. That table also gives the operations resulting from each instruction, where "reg" is the register named as the first operand, and "op2" is the register or array element which is the second operand. Several examples are given in Section II.B.9.c, below.

The use of this format has the advantage that it can be used with versions of the preparation program which produce the mechanism dependent subroutines in machine code. (This version is used as SAPRC for research because it is much faster to prepare mechanisms, since it is not necessary to relink the model simulation program for each mechanism. Thus, the

Table 2. Summary of Instruction Codes and Operands for Inserted Instructions for the Machine Language (".#Dx") Format

Operations:

Code	Description	Operation	Comments
C	Clear	reg ← 0	No second operand
N	Negate	reg ← -reg	No second operand
L	Load	reg ← op2	
A	Add	reg ← reg + op2	
S	Subtract	reg ← reg - op2	
M	Multiply	reg ← reg x op2	
D	Divide	reg ← reg / op2	
E	Store	op2 ← reg	Second operand not a register

Operands:

Code	Description
0	Register 0 (first or second operand)
1	Register 1 (first or second operand)
2	Register 2 (first or second operand)
C'species name'	Active or constant species (a)
K'reaction label'	Rate constant (a)
#'coefficient'	Variable coefficient (a)

[a] Legal for second operand only.

inserted instructions used in the mechanism given in the example application listed in Appendix A are given in this format.) However, this format is not as well adapted to this and other versions of the preparation program which produce the mechanism dependent subroutines in FORTRAN, since a rather awkward and inefficient appearing FORTRAN code is produced. For example, the set of instructions

```
L1#'1',S1#'A',E1#'1MINUSA'
```

which subtracts the variable coefficient "A" from the constant coefficient 1 and stores the result in the coefficient called "1MINUSA", might produce the following FORTRAN code:

```
R1 = CO( 421)
R1 = R1 - CO(2)
CO(  3) = R1
```

if the coefficient numbers for "A", "1MINUSA", and "1" are 2, 3, and 421, respectively. The corresponding operation in the pseudo-FORTRAN syntax ("'1MINUSA'='1'-'A'", see below) would result in the code

```
CO(  3) = CO(421)-CO(  2)
```

which appears to be significantly more efficient. However, it should be noted that if the subroutine is compiled using an optimizing compiler, the resulting object code would probably not be significantly different than if the pseudo-FORTRAN format were employed.

b. Inserted Instructions in Pseudo-FORTRAN Format (.INS)

When the inserted instruction code ".INS INIT" or ".INS DIFF" are used, the inserted instructions codes are expected to be given in the second format, where the instructions consist of FORTRAN-like statements. Only one instruction is permitted per line, and the instruction must begin in the first column. As with FORTRAN statements, the operand whose value is being modified is given first, followed by an "=", and then the various operands and operations. Any operation which is legal in FORTRAN for real variables can be employed, but the most frequently used operations would be "+", "-", "*", and "/", for add, subtract, multiply, and divide. Operands can be any legal FORTRAN real variable name (except for "A", "C", "CO", "S", "RK", "R", "FR", or "LR", which are used for the names of the internal arrays), or special symbols enclosed in single or double quotes or brackets, which are decoded into the appropriate array referencing a species concentration, coefficient value, or reaction rate constant as shown in Table 3.

Note that any symbols in the inserted instruction which are not recognized as being in the format given in Table 3 (i.e., as indicated by the use of the single quote, double quote, or bracket characters) are copied directly to the output code, without any error checking. Thus the only processing done on these pseudo-FORTRAN instructions is to convert the characters surrounded by the quotes to the appropriate reference to the internal array of concentrations, coefficient values, or rate constants, and to start the instruction in column 6, as required by FORTRAN.

c. <u>Examples of Inserted Instruction Input</u>

In this section, several examples of inserted instruction input are given to illustrate its usage. These examples are given in both formats, except for the last, where only the pseudo-FORTRAN format can be employed.

<u>Example 1: Deriving One Coefficient from Another</u>. There are frequently applications where one variable stoichiometric coefficient is completely determined by another. For example, suppose a species can react to form either one of two different products, and calculations where the branching ratio is varied is desired. In order to do this, the stoichiometric coefficients for these two products must be given as variable coefficients, which we will designate "Y1" and "Y2". Note that Y1 and Y2 are not independent, since they must sum up to 1. If no inserted instructions are given to enforce this dependence, it is necessary to give both values as input when carrying out the calculation, and to be sure that they sum up to one. This relationship can automatically be enforced by giving the inserted instructions forcing Y2 to always be 1-Y1, so that Y1 is the only coefficient that need be specified. Since Y1 and Y2 are constant, the calculation of Y2 need be carried out only once, in the initialization phase of the calculation. The inserted instructions which

Table 3. Summary of Special Operand Symbols Used for Inserted Pseudo-FORTRAN Inserted Instructions (".INS" Format)

Operand Symbol	Decoded As
"C species name"	Active or constant species
"K reaction label"	Rate constant
"# coef name"	Variable or constant coefficient
[species name]	Active or constant species
'coef name'	Variable or constant coefficient

can thus be included to take care of this are as follows:

Pseudo-Machine Format	Pseudo-FORTRAN Format
.#C1	.INS INIT
L1#'1',S1#'Y1',E1#'Y2'	'Y2'=1.0-'Y1'
	or
	"# Y2"=1.0-"# Y1"

This causes the instructions to calculate Y2 from 1-Y1 to be included in the CONSTR subroutine.

(Note that the pseudo-FORTRAN instructions differ from the pseudo-machine language instructions in that the latter uses a constant coefficient "1", while the former uses that constant as a local constant in the FORTRAN subroutine. They would have been more strictly equivalent if the pseudo-FORTRAN instructions were

```
'Y2'='1'-'Y1'          or     "# Y2"="# 1"-"# Y1"
```

The use of constant coefficients as local FORTRAN constants rather than references to an element in the coefficient array may be somewhat more efficient, depending on the compiler. However, the net effect is the same.)

Example 2: Initial Concentrations Derived from Others. In some models for chamber effects in environmental chambers, it is assumed that a certain fraction of the initially injected NO2 is converted to HONO during the injection process, or by interaction with the chamber walls before the irradiation. If the fraction of initially present NO2 which is converted to HONO is given in the coefficient "YHONO", this can be implemented by the following set of instructions, which are inserted in the initialization code. Note that both NO2 and HONO are active species.

Pseudo-Machine Format	Pseudo-FORTRAN Format
.#C1	.INS INIT
L1C'NO2',M1#'YHONO',E1C'HONO'	[HONO]='YHONO'*[NO2]
L1C'NO2',S1C'HONO',E1C'NO2'	[NO2]=[NO2]-[HONO]

these instructions result in the initial HONO concentration being set at the initial NO2 times YHONO, and the initial NO2 being reduced by that amount.

(Note that the grouping of the pseudo-machine instructions into the two records shown in the above example is entirely arbitrary. The use of commas to separate the instructions, when there are more than one per record, is also arbitrary, but a recommended convention.)

<u>Example 3: Coefficients which Depend on Concentrations</u>. For some lumped mechanisms, product yield coefficients might be represented as depending on reactant concentrations. Assume that the reactions

```
A = S
S + B = P1
S + C = P2
```

are such that the first reaction is always rate determining, and the rate constant for the second reaction is 3.4 times larger than the first. These reactions might be approximated as

```
A = #Y1 P1 + #Y2 P2
```

where

```
Y1 = [B]/([B]+3.4[C])
```

and

Y2 = 3.4[C]/([B]+3.4[C]).

Inserted instructions could be used to calculate the values of the coefficients Y1 and Y2 from the concentrations of B and C. If B and C are both constant, then the inserted instructions could be added in the initialization code. If one or both are active, then the instructions must be carried out at each step of the calculation. This can be implemented as follows:

Pseudo-Machine Format	Pseudo-FORTRAN Format
.#D1	.INS DIFF
L1C'C',M1#'3.4',L21,A1C'B'	X=3.4*[C]+[B]
D21,E2#'Y2'	'Y1'=[B]/X
L2C'B',D21,E2#'Y1'	'Y2'=3.4*[C]/X

Example 4. Conditional Tests. In some special cases, it may be desired to modify coefficients or rate constants depending on values of concentrations of active species. Some limited capability is provided by the use of the pseudo-FORTRAN format of the inserted instructions, since any type of valid FORTRAN statement can be processed as inserted instructions, including IF statements. For example, suppose tests showed that the simulation proceeded more rapidly if the rate constant for a reaction labeled "R33" were set to zero if the concentration of a species "X" fell below 1.0E-5 ppm. This could be implemented by the following instructions:

```
.INS DIFF
IF ([X].LT.1.0E-5) "K R33"=0.0
```

This capability is not not provided by the pseudo-machine language format of inserted instructions.

10. Null Records

The null record, "." is used to indicate the end of one of the commands (e.g., .RXN), or the end of one of the conditional commands (FOR or IF). Its use with conditional commands are discussed in Section

II.B.12, below. All noncommand input following it, up to the next command record, are ignored by the program.

11. .END Command

After all of the reactions, species, and other characteristics of the mechanism have been specified, the .END command must be included to tell the program to terminate input and begin processing the mechanism. Any data in the input file following the .END command is ignored, so normally the .END command should be the last record in the input file. This is required input. If an end-of-file on the input file is encountered, the mechanism is not prepared, though the input data are processed for other errors.

12. Conditional Input

The preparation program has several features for conditional processing of the model specification input. This is implemented by two commands or subcommands, FOR and IF. The FOR option for conditional input is used to avoid duplicate processing of input in the model which might be in several different files which are used for redirected input by the @filename command (see next section). For example, if the chemical mechanism is split up into several modules giving reactions of different organics (some of these organics may have several intermediates in common), the reactions and other specifications for those intermediates can be included in those files with the FOR condition in effect, which results in those reactions and other input being processed only the first time they are encountered, with the subsequent input for them being ignored.

Conditional input under the FOR option can be specified in two ways:

```
. FOR label
```

or

```
.cmd FOR label
```

where the "." must be in the first column, "label" is a 1-8 alphanumeric character name, and ".cmd" is any of the commands listed in Table 1 except for .UNITS, .INS, or .PHOT. With the first format, the input which is defined as conditional are all the records following that command until a null record (a lone "." or a ". FOR label", or a ". IF label" record) is encountered. This permits a group of commands (such as perhaps .RXN to specify reactions, .STS to indicate which reactants in those reactions are in steady state, .COE to give named coefficient values, etc.) to be included in one conditional input set. With the second format, the conditional input is only those records associated by that command (i.e., the conditional input ends when another command record is encountered).

When "FOR" is encountered by the program, the records in that conditional section are taken as input only if no previous "FOR" command with the same label has been encountered. If a FOR command with the same label was previously processed, the program ignores all records in the conditional section. The assumption here is that the data in that section has already been input. Note that the label names must match identically. A ". FOR" command without a label is ignored.

The conditional "IF" is useful when a limited number of certain reactions or other data are to be sometimes used and sometimes not in a model. The syntax for the "IF" command is completely analogous to the "FOR" command, i.e.,

```
. IF label
```

or

```
.cmd IF label
```

and the rules determining the input which is conditional are the same. The difference is that in this case, the input affected by the conditional command are processed only if an input record consisting of

```
. label
```

(where "label" is the same as that used in the corresponding "IF" subcommand, and a single space is between the "." and the label. Since the commands are processed sequentially, this record (which indicates that the "condition" referenced by "label" is "true") must be input before the IF conditional input referencing is processed, or the conditional data will not be processed. Thus, to include the conditional input, a null command line with the label would be placed somewhere near the beginning of the

PRP file, and the appropriate section(s) of data delineated with "IF" command(s) with the matching label.

13. Redirected Input

Input can be taken from another file at any point in the input file. This is done by including a record with a "@" as the first character, followed by (with no blanks) a legal file name recognized by the system. When such a record is encountered, subsequent input is taken from the referenced file until the end of file is reached. The input format of the data in the secondary input file is exactly the same as from the main preparation program. Indeed, the modules of the preparation program which process the input data do not "know" whether the input is coming from the primary input (.PRP) file or a secondary input file referenced by it, since the processing of the input and the opening and closing of the secondary files is handled by an input subroutine which passes the records from the currently open input file to the model preparation modules. Once the end of the secondary file is reached, that file is closed and input resumes from the next record of the included file. This secondary input file is called an "included" file.

Included files can reference other included files, but no more than three input files can be open at once. This means that while included files referenced by the primary (.PRP) input file can reference other included files, those other included files cannot themselves reference included files.

An example of the use of included files is the manner in which the input data defining reaction mechanisms (as developed at SAPRC) are organized. The reaction mechanisms are organized into modules, with separate files for different types of reactions. (These are by convention named .RXN files, but this nomenclature is not required by the program since the full file name must be given in the "@filename" command.) Thus, for example, the .PRP file for a mechanism including only the reactions of acetaldehyde and formaldehyde might consist of the following:

```
.REACTIONS OF LOWER ALDEHYDES
TEMP 300.
```

```
.UNITS=PPM
@NOXAIR.RXN
@RO2.RXN
@HCHO.RXN
@CCHO.RXN
.END
```

where the @NOXAIR.RXN, @RO2.RXN, etc., cause input to be taken from files giving the reactions and other input for the specific groups of species in the mechanism. A mechanism for a full airshed model would obviously include input files for many additional classes of organics.

Note that the "@filename" is not affected by conditional input commands. The file input is redirected even if the "@filename" command is processed in the middle of an IF or FOR group where preparation program commands are being bypassed. This is the case because as indicated above the processing of the redirected input command occurs prior to passing the input records to the portion of the model preparation program which processes the chemical mechanism data. That portion of the program which processes the mechanism input is the same portion that processes the conditional input commands.

14. Special Commands

Special commands are commands given to the file input processor, which controls input going to model preparation program itself. These commands are indicated by a "@" in the first column of the record. One example of special commands is the "@filename" command, which is used for redirected input as described in the previous section. Other special commands which are currently implemented are the "@#SHOW", "@#NSHOW", and "@#END" commands. The "@#SHOW" (which can be abbreviated "@#SH") causes all input records from the .PRP (or its included input files) to be displayed at the terminal as they are being processed. The "@#NSHOW" (abbreviated "@#NS") cancels the effect of "@#SHOW" and reverts the program to the default state where such input is not displayed. The "@#END" command causes all input files to be closed and an end-of-file condition to occur. It is not normally used in the preparation program input. Note that the processing of these special commands are not affected by the conditional input commands, since they are processed by the input interface subroutine which passes input to the program module that processes conditional input.

C. Species Types

Six different types of species are recognized by the preparation program. These are active, build-up, steady state, constant, saved dummy, and deleted dummy. By default, all species which undergo at least one reaction are active; all species which are formed in a reaction but do not react themselves are build-up; there are no dummy species; and the species will be ordered in the mechanism in the order they are encountered when the reaction input is processed. However, these defaults can be overridden by species specification input. The format of the species specification input was described in Section II.B.4, above. The meanings and limitations of the various types of species recognized by this program are described in this section.

1. Active Species (.ACT)

Active species are those which are included in the integration as normal species without any special treatment. This is the default type for any species which participate in a reaction. It should be noted that the more active species there are, the longer the integrations will take. The integration program has a limitation on how many active species it can handle. If a species undergoes a reaction in the mechanism, it does not have to be declared explicitly as active using an .ACT command, since this would be the case by default. However, the .ACT command can be used to assign default initial concentrations, molecular weights, numbers of carbons and nitrogens, and to determine the order in which the active species are listed in the output files defining the mechanism for the airshed modeling program. Note, however, to change the species ordering the .ACT command, listing those species should be given before any reactions are processed, or the species encountered in those reactions will occur first in the final ordering.

Another case when the .ACT command may be needed is to declare as active species those which are formed but do not undergo any reaction. This forces these species to be processed in the same way as all other active species. Otherwise, species which do not react are classified as "build-up", and their rates of formation are calculated by a separate subroutine, which may or may not be utilized by the airshed model software.

2. Build-up Species (.BLD)

Build-up species are formed but do not react. Since the calculated concentrations of such species do not affect any other predictions of the airshed model, the code for calculating their rates of formation are prepared in a separate model-dependent subroutine module to permit for separate processing (or not processing) by the airshed model (see Section IV.A.3). This is the default type for species which are formed but do not react, and which are not declared explicitly as active using the .ACT command. If a species is declared as build-up using a ".BLD" command, and it in fact undergoes a reaction in this model, a warning message is given in the preparation program listing output (.PRO) FILE, and its type is changed to active. Thus the only purpose of ".BLD" commands are to assign default parameters (initial concentrations, molecular weights, etc.) to such species, or for determining the order in which build-up species are organized in the airshed model.

3. Constant Species (.CON)

As their name implies, constant species are constant throughout the model simulation. If a constant species appears as a product in a reaction list, it is ignored (i.e., treated as if it didn't appear in that reaction at all). No species are constant by default, so all constant species must be explicitly specified as such with ".CON" commands. This command can also be used to give the default values for their concentrations, their molecular weights, etc., as discussed above. These default values are passed to the airshed model program as part of the mechanism's .MOD file, where the airshed program can use or override the default as appropriate.

4. Steady-State Species (.STS)

Steady-state species are treated in pseudo-equilibrium at each step in the integration, thus avoiding the computational or memory allocation overhead required for active species. This option can also significantly reduce the mathematical "stiffness" of the differential equations in the airshed model if applied to the appropriate set of compounds, which can also significantly improve computational time. No species are in steady state by default, so all such species must be explicitly declared as such with ".STS" commands. The preparation program software automatically produces the computer code to calculate the steady-state concentrations of these species, and the rates of the reactions in which they undergo. However, for this processing to be successful, there are several restrictions regarding which species can be declared as in steady state. The restrictions imposed by the software are as follows:

(1) Steady-state species must be both formed and undergo reaction. If no reaction forms a species designated as steady-state, it is changed to constant, with a default concentration of zero. If it does not react but is formed, it is changed to build-up. A warning message is included in the mechanism listing output (.PRO) file, but the processing of the mechanism is not aborted.

(2) No steady-state species can react with itself or with any other steady-state species. If a reaction is encountered where this rule is violated, an error message is given to the terminal and the mechanism listing output file, and the mechanism preparation fails.

(3) "Cyclic" systems, such as

$$A + S1 = S2 + D$$
$$B + S2 = S3$$
$$C + S3 = S1$$

where S1, S2, and S3 are steady-state species and A, B, C, and D, are active species, are not allowed. Another example of a "cyclic" system is a reaction where steady-state species react to regenerate itself, though there is a special exception where this type of reaction is allowed (discussed below). If a "cyclic" condition for any subset of the steady-state species is detected, an error message is given and the model preparation fails. (In photochemical smog models, this problem can frequently be avoided by not having HO2 radicals in steady state.)

The last requirement is imposed because of the way which the preparation program has the model-dependent subroutines calculate the concentrations of steady-state species. The procedure used requires that the steady-state species be ordered such that the concentration of the first steady-state species in the ordering can be calculated independently of the concentrations of any other steady-state species, and the n'th species in the ordering depends only on concentrations of species preceding it in the ordering. Thus, this determines the ordering in which the concentrations of the steady-state species are calculated. (This ordering only affects the code in the model-dependent subroutines; it does not affect the ordering of how the steady-state species are stored in the airshed model, or how they are listed in the model documentation output.) If a "cyclic" condition occurs, it is not possible to order the species in this way.

Because of the way in which steady-state species are processed by the preparation program, special treatment is required for reactions where a steady-state species reacts to regenerate itself in a single reaction. An example of such a reaction might be a reaction where a steady-state species acts as a catalyst, e.g.,

$$S + A = S + B$$

where S is the steady-state species. The software will process such a reaction ONLY if (1) the steady-state species is the first species on the reactant list, (2) the steady-state species is also the first species on the product list, and (3) the steady-state species is formed as a product without a stoichiometric coefficient. This special format tells the software that the rate of the reaction has no effect on the concentration of the steady-state species, so the reaction can be ignored when determining the ordering for the steady-state species for the purpose of calculating their concentrations. It is possible to modify the preparation program to make the format requirements for such reactions less restrictive, but this has not yet been done in this version.

In addition to the requirements imposed by the model preparation software, it is obviously also important that the steady-state approximation be appropriate for any species declared as in steady state. That is, it must react with a lifetime far shorter than the time constant for the overall change in reactant concentrations. Otherwise, the results of the computation will be meaningless, or the computation may fail. There is no way the preparation program can determine if this is the case, so appropriate judgment must be exercised. If the user is uncertain as to whether the steady-state approximation is valid, he can prepare versions of the mechanism with and without the species being in steady state to see if it has any effect on the results of a representative variety of model simulation problems.

5. Dummy Species

Dummy species are species which neither react nor are formed in any reaction. This can occur if a species is specified as being of a certain type in a species specification command, but there is no reaction in which it is formed or reacts. If a dummy species is specified in a .ACT, .BLD, .CON, or .STS command, the preparation program will list it in the species listed as a "Dummy Species Deleted", and it will be deleted from the mechanism before the mechanism is prepared for use in the airshed model. A warning message may also be given, depending on the type of species it was declared to be. This is not regarded as an error condition by the preparation program, but usually indicates that there was some type of input error, usually the species name being misspelled in the species specification input or in the reaction.

In some cases it may be desired to include species in the model for which no reactions are given. This can be done by using the ".DUM" command to declare them as "saved" dummy species. The ".DUM" command can also be used to assign their default initial concentrations, molecular weights, etc., as is the case with the other species types (except steady state). The names, initial concentrations, molecular weights, and numbers of carbons and nitrogens for these species, given in the ".DUM" command, are passed to the airshed modeling program without any additional processing. If a species is declared as dummy, but it actually reacts or is formed in a reaction, then a warning message is given, and it is changed to active or build-up only, whichever is appropriate.

III. EMISSIONS PROCESSING PROGRAMS

The emissions processing system described in this section consists of a group of interrelated programs and data sets which can be used to convert emissions data from the form in which they generally occur in emissions inventories into a form which can be directly utilized by airshed modeling programs. It is intended to be used with airshed models which incorporate chemical mechanisms prepared using the preparation program, as described in the previous section. The software developed for this purpose consist of the PREPEMIT and the LUMPEMIT programs. The PREPEMIT program is used to prepare emissions assignments for a specific chemical mechanism and lumping scheme, and to derive mechanistic parameters for the lumped mechanism which best fit the distribution of species in a representative emissions profile. The LUMPEMIT program is a simplified example of a program which pre-processes emissions data for use as input into airshed models, and employs the output of the PREPEMIT program to determine the lumped species assignments for the various emissions classes.

The use of this emissions processing system is discussed in this section. To utilize this system, it is necessary to have an overall understanding of the various emissions classification schemes and data sets employed, and an overview of these is given in Section III.A. In Section III.B, the operation and use of the PREPEMIT emissions preparation program and of the LUMPEMIT emissions processing programs are discussed. In Section III.C the formats of the various types of user-maintained input data sets required by these programs are described. Finally, in Section III.D, special considerations involved in the use of "lumped groups," which are special species in the mechanism whose parameters are adjusted based on the emitted species which they represent, are discussed. The internal software components of these programs are described in Section IV, and specific examples of their applications are given in Section V.

A. Overview of Emissions Classifications and Data Sets Employed

In order to understand the operation of this emissions processing system, it is necessary to understand how the emissions are classified before, during, and after the operation of this software, and the purpose and concepts behind the various types of data sets that are employed. These classification systems and data sets are summarized briefly in this section, along with a general indication of how they are utilized by the emissions preparation (PREPEMIT) and the emissions pre-processing (e.g., LUMPEMIT) programs.

Species or Class. In the context of this discussion, the terms "species" or "class" refers to a group of one or more emitted chemical compounds which are organized according to some type of classification scheme. Three different types of classification schemes for species are employed in this system, the "SAROAD" classification used in the emissions inventories, the "lumped model species" classification used in the chemical mechanisms in the airshed models, and a "detailed model species" classification which is used internally by the PREPEMIT program and some of its input and internal data sets. Each of these are discussed further below.

SAROAD Species or SAROAD Class consist of the classes used in the SAROAD system of classifications of emitted chemicals. It is assumed that the emissions data used as the input into the emissions processing programs discussed in this document are initially organized according to this system. This SAROAD system has evolved over the years for the purpose of classifying emitted chemicals, and is widely employed for emissions inventories, including those maintained by the EPA and the CARB. It employees 5-digit numbers to identify emitted chemicals or groups of chemicals; these numbers are called the SAROAD numbers in the documentation to this software. Each SAROAD number can refer either to an individual chemical (e.g., 43235 for n-nonane and 43310 for 2-methoxy ethanol), to a mixture of similar chemicals which may or may not have similar reactivity (e.g., 43122 for "isomers of pentane," and 98049 for "C9 aromatics") or to a less well characterized mixture of chemicals (e.g., 45101 for "naphtha" and 98098 for "paraffin bond"). Although all emissions inventories employing the SAROAD scheme are supposed to have a consistent numbering scheme and there is supposed to be a one-to-one correspondence between SAROAD classes and chemical species, there are some inconsistencies between the scheme currently being employed by the CARB and the EPA, and (particularly in the EPA scheme), there are numerous examples where more than one SAROAD number can refer to the same chemical or group of chemicals. A more serious problem is that a large number of SAROAD classes refer to groups of compounds which, though isomeric and chemically similar in many respects, may have significantly different atmospheric reactivity, and assumptions have to be made concerning the distribution of specific compounds they represent. For example "isomers of butene (SAROAD no. 43120, an important component of many emissions profiles), could include 1-butene, the 2-butenes, or isobutene, each of which are quite different in atmospheric reactivity. The emissions assignment data sets included with this software include listings of the SAROAD classes employed in the emissions data bases we have received from the CARB and the EPA, and comments indicating the inconsistencies between them. (See Appendix A for a listing of the computer-readable data sets included with this software). However, except for those 29 classes which refer to different chemicals on each data base, the emissions processing software discussed in this document can be employed with either of these data bases.

Emissions Input Data. In the context of this documentation, the term "emissions input" will refer to emissions data organized according to the SAROAD classification scheme, with the amount of each SAROAD class emitted being given in terms of mass per unit time. (The software has no specific requirement as to which mass or time units are employed, though in general an appropriate conversion factor will need to be input into the emissions processing program to assure the output is in the units utilized by the airshed model. If no conversion factor is input, the program assumes that the mass units are grams, and the time units are the same as utilized by the airshed modeling program.) The emissions input data sets thus consist of sets of pairs of numbers, the SAROAD number and the mass per unit time emissions of that SAROAD class. The exact format of these data sets are discussed in Section III.C.1.

Detailed Model Species consist of the chemical species or groups of chemical species of similar reactivity for which a chemical mechanism or model has been derived. This classification scheme is intended to represent the maximum level of chemical detail available in the chemical model data base, which in general is less than the level of chemical detail contained in the SAROAD classification scheme (excluding those SAROAD classes whose chemical compositions are poorly characterized), but is much greater than the level of chemical detail represented by the lumped model species employed in the airshed model itself.

The detailed model species include not only species whose reactions are incorporated explicitly into the airshed model (i.e., the "explicit model species," discussed below), but also a much larger set of species whose kinetic and mechanistic parameters are given in the mechanistic parameter files which are among the input to the PREPEMIT program. The PREPEMIT program uses the parameters for these detailed model species to derive the kinetic and mechanistic parameters for those lumped model species whose parameters depend on the mixture they are representing. The detailed model species are identified by an 8-character (maximum) species or group names, or by an 8-character mechanism code and a set of parameters giving the OH radical rate constant as a function of temperature. Some of these species refer to single chemical compounds, while others refer to groups of compounds which are assumed to have the same reactivity. Examples of such species in the present chemical model data base are HCHO (formaldehyde), N-C8 (n-octane), BR-C12 (branched C_{12} alkanes), C6-OLE2 (C_6 alkenes with internal double bonds), etc. If OH radical kinetic parameters are included with the species designation, then the 8-character code is used to determine the mechanistic parameters, with the kinetic parameters for the OH reaction being given explicitly. For example, in the current chemical model, the mechanistic parameters for all haloalkanes are assumed to be the same (and identified by the code "CL-ALK"), with the appropriate rate constant being specified separately for each individual compound.

<u>Detailed Model Species Composition Sets</u> consist of the emissions data categorized in terms of molar emissions of detailed model species. These are used internally by the PREPEMIT program, and are not output as separate data files in the normal operation of the system. The PREPEMIT program uses the data in the SAROAD.PRM data set (discussed below) to convert mass emissions of the SAROAD classes in the representative profile to a detailed model species composition set for that profile. The program then uses these data to determine the mechanistic parameters for the lumped classes, and for determining how the input emissions data are to be apportioned among the lumped model species, based on the input data in the lumping control file (also discussed below).

The SAROAD.PRM Data Set is the computer data file required as input to the PREPEMIT program which contains the assignments of detailed model species for the SAROAD classes. Since some SAROAD classes consist of mixtures of chemicals of different reactivity, in general more than one detailed model species may be assigned to a given SAROAD class. For each SAROAD class, this data set gives the detailed model species assigned to it, and the weight fractions and molecular weights assumed for each, as well as code numbers (not used by PREPEMIT) giving uncertainty codes and footnotes about the assignments, which are explained in "comment" records at the beginning of the data set. As indicated above, the detailed model species is identified by an 8-character (maximum) specifying a type of chemical mechanism, and in some cases also kinetic parameters for the OH radical reaction, if those associated with the 8-character code is not appropriate to this SAROAD class. The format of this data set is described in Section III.C.2.

Lumped Model Species consist of the chemical species and lumped classes which are actually used in chemical mechanisms in the airshed model to represent the emitted species. These are related to the detailed model species based on the data in the "lumping control files," discussed below. The lumped model species represent the level of chemical detail incorporated in the airshed model, where computer software and time constraints require much lower levels of detail than would be represented by the set of detailed model species. All these species must be included among the species input into the preparation program when preparing the mechanism to be used in the airshed model. As discussed below, there are two types of lumped model species: "explicit" model species or "lumped groups." Lumped model species are identified by a four-character (maximum) code, examples in the current SAPRC mechanism being HCHO (for formaldehyde), MEK (used for C_{4+} ketones), and AARZ (used for the second most rapidly reacting lumped class of alkanes or aromatics), etc.

Explicit Model Species are lumped model species which are represented explicitly in the mechanism and whose kinetic and mechanistic parameters are determined by the preparation program input and are not modified based on the emissions profiles. This can include species which represent only a single compound (and thus in general may be the same as a detailed model species), and also species which are used to represent a group of compounds based on the "surrogate species" approach. (The "surrogate species" approach is a lumping approach where the reactions of a number of molecules are represented by the reactions of a "typical" or representative molecule whose reactions are assumed to have the same effect on the system.)

Lumped groups are special lumped model species which have variable mechanistic and kinetic parameters whose values are derived by the PREPEMIT program based on the specific set of compounds they are representing, as determined by the data in the lumping control files (see below), and the composition of the "representative" emissions profile input into PREPEMIT. These species are used to represent compounds by what we designate the "lumped parameter" approach, where the mechanistic and kinetic parameters (i.e., rate constants and product yield stoichiometric coefficients) for the lumped species are weighed averages of those for the sets of compounds they represent. This approach is designed to permit maximum accuracy and flexibility in representing in the model a variety of mixtures of compounds with only a limited number of model species.

The Lumping Control Files are computer data sets which can be used as input to the PREPEMIT program to tell it (1) what are the lumped model species being used, (2) how to relate detailed model species to the lumped model species, and (3) which data files it should read to get the kinetic and mechanistic parameters for the detailed model species which will be associated with the lumped groups. For example, input in this data set determines how many lumped groups will be used for various types of chemicals (e.g., alkanes, aromatics, and alkenes, and the minimum and maximum rate constant for the compounds to be associated with each lumped class. The PREPEMIT program then apportions the various chemicals of these types into the different lumped groups, based on their OH radical rate constant (obtained from the kinetic and mechanistic parameter data files, or by the detailed model species specification in SAROAD.PRM). For detailed model species for which kinetic and mechanistic data are not read in, the data in the lumping control file is used to determine which "explicit" lumped model species is to be used to represent it. The format of the data in this file is discussed in Section III.C.3, and considerations regarding relationships between the data in the lumping control file and the data used to specify the chemical mechanism in the preparation program input file are discussed in Section III.D.

Several representative lumping control files are included with this software, but these can be modified by the user if differing lumping approximations are desired. More than one lumping control file can be used for a given chemical mechanism, provided that they use the same explicit lumped model species and the same numbers and types of lumped classes. However, all emissions input to a given airshed model run must be processed using the same lumping control file, since using different lumping control files in effect means using a different chemical mechanism.

<u>The Kinetic and Mechanistic Parameter Files</u> give the kinetic and mechanistic parameters for detailed model species which are not explicitly represented in the mechanism, but which are represented in the mechanism by inclusion in one of the lumped classes. In the present chemical model, these are primarily aromatics, alkenes, alkanes, and various miscellaneous compounds (e.g., alcohols, ethers, esters, etc.) which are generally lumped with alkanes. There are three possible formats for the data in these files, which are appropriate for different types of compounds, and these are discussed in Section III.C.4.

<u>The Lumped Species Parameter File</u> is an output of the PREPEMIT emissions preparation program. This file is intended to be read by the airshed modeling program as part of the mechanism initialization process, before any of the emissions data are read in. (An example of airshed model initialization subroutine, RDPRM is included with this software for this purpose.) This file contains (1) the list of all of the lumped model species being used to represent emissions, in the same order as they are listed in the lumped SAROAD assignment file, and thus in the same order that they would be output by the LUMPEMIT program; (2) the weighed average number of carbons and molecular weights for each of these lumped model species, based on the distribution of compounds each represents in the typical emissions profile input to PREPEMIT; and (3) the weighed average kinetic and mechanistic parameters for the lumped classes in the mechanism, derived based on the sets of compounds in the input profile that each represents. (This file can also optionally contain the compositions of the typical emissions profile in terms of the lumped model species.

The users of the airshed modeling program can optionally use these data as default compositions for emissions where no composition data are available, though this might not always be appropriate, and the specific software implementing this has not been prepared.) The format of this and other output data sets produced by PREPEMIT is described in Section IV.B.3.

The Lumped SAROAD Assignments File is an additional output of the PREPEMIT emissions preparation program. This file is used to inform LUMPEMIT (or other emissions pre-processing programs) how to convert emissions input data (i.e., mass emissions of SAROAD species) into processed emissions data (molar emissions of lumped model species) for use by the airshed modeling program. It contains (1) a list of all of the lumped model species being used to represent emissions, in the order they will be expected by the airshed modeling program (based on the data it reads from the lumped parameter file, discussed above), and (2) lumped model species assignments for all the SAROAD classes, and the number of moles of lumped model species corresponding to each gram of emissions into the SAROAD class. Note that since some SAROAD classes refer to mixtures of compounds of different reactivity, some SAROAD classes are apportioned between two or more lumped model species, and this information is also given in the lumped SAROAD assignment file. The contents of this file is determined primarily by the data in the SAROAD.PRM emissions assignment file and the lumping control file, but in some cases it may also be affected by the composition of the representative emissions profile used as input to the PREPEMIT program. (Specifically, this is the case if there are lumped classes where OH rate constant weighing is used in the averaging. This is an option which can be specified by the lumping control file. In such cases, the contribution of a SAROAD class to the lumped class would depend on the magnitude of its OH radical rate constant relative to that assigned to the lumped class, where the latter may have been derived based on the representative emissions profile. See Section III.D.2 for a discussion of rate constant weighed averaging.) The same lumped SAROAD assignment file must be used in processing all emissions input used in a given airshed model run.

Processed Emissions Data refers to emissions data which are organized in a form which can be directly utilized by the airshed modeling program.

These data give the emissions in terms of moles (or something proportional to moles, such as ppm in a given volume of air at a given temperature and pressure) of lumped model species per unit time. These data are output of the emissions pre-processing programs, of which LUMPEMIT is an example. The units of the molar emissions can be controlled by giving the LUMPEMIT program an appropriate conversion factor (see Section III.B.2); if no conversion factor is used, and if the emissions input data are in units of grams per unit time, then the processed emissions data are output in units of moles per the same time unit. Processed emissions data can be either in the form of single arrays of molar emissions amounts given in a standard order determined by the input to PREPEMIT (where the emissions pre-processing and airshed modeling programs determine the ordering of the lumped species from the lumped species parameter file output by PREPEMIT), or in pairs of arrays consisting of the four-character code word and the corresponding molar emissions amount. In general, there will be many sets of such arrays used as input into an airshed model simulation, representing the full temporal and spacial variability of emissions into the modeling domain.

B. Operations of the PREPEMIT and the LUMPEMIT Programs

1. PREPEMIT Program

a. Operation of the Program

The PREPEMIT program operates in three separate steps, which are shown diagramatically in Figure 3. These three steps are as follows:

(1) In the first step, the program reads an emissions input data set giving the typical or representative profile of the emissions into the modeling region of interest. The program prompts the user for the name of a data file containing such data. It then converts these data, given initially in terms of mass emissions of the various SAROAD classes, into molar emissions of detailed model species, which are required for subsequent processing. The program uses the data in the SAROAD.PRM data sets to determine how the detailed model species are assigned to the various SAROAD classes, and to convert mass emissions of SAROAD classes to molar emissions of detailed model species. These "detailed model species composition" data are not output by the program, but are stored internally for the subsequent steps.

```
                                              +--------------------------------+
                                              !    Representative Emissions    !
                                              !   Profile (Mass Emissions of   !
                                              !        SAROAD Classes)         !
                                              +--------------------------------+
                                                              !
                                                              V
                          +-----------------+        +-------------+
                          !   SAROAD.PRM:   !        !             !
                          ! Detailed Model  !        !             !
               +---------! Species Assign- !-------->!   STEP I    !
               !          !    ments for    !        !             !
               !          ! SAROAD Classes  !        !             !
               !          +-----------------+        +-------------+
               !                                             !
               !                                             V
               !                               +--------------------------------+
               !                               ! Detailed Species Compositions !
               !                               !  of Representative Emissions  !
               !                               +--------------------------------+
               !                                              !
               !                                              V
               !          +-----------------+          +-------------+
               !          ! Lumping Control !          !             !
               !    +----!      File       !--------->!             !
               !    !     +-----------------+          !             !
               !    !                                  !   STEP II   !
               !    !     +-----------------+          !             !
               !    !     ! Detailed Model  !--------->!             !
               !    ! +--!     Species     !          +-------------+
               !    ! !   !   Parameters    !            !         !
               !    ! !   +-----------------+            !         !
               V    V V                                  V         V
+-------------+                              +----------------+ +----------------+
!             !    (lumped group             ! Lumped Species ! ! Lumped Species !
!  STEP III   !<-----------------------------! Parameters for ! ! Composition of !
!             !    rate constants)           ! Rep. Emissions ! ! Rep. Emissions !
!             !                              +----------------+ +----------------+
+-------------+
       !
       V
+-----------------------------+
! Lumped Species Assignments !
!     For SAROAD Classes      !
+-----------------------------+
```

Figure 3. Diagram of the relationships between the three steps of the operation of the PREPEMIT program and data sets employed.

(2) In the second step, the program uses the data in a "lumping control file" to determine how to convert detailed model species to lumped model species, and to obtain the kinetic and mechanistic parameters for those detailed model species which are to be represented by lumped groups. The program prompts the user for the name of the lumping control file to use for this purpose. Once the data from this file is read in, it then uses these data to (a) convert the molar emissions of the detailed model species, as determined by the first step of the program, into molar emissions of the explicit model species and the lumped classes, and (b) determines the kinetic and mechanistic parameters of the lumped groups which correspond to the sets of detailed model species which they represent. The program then outputs these data into the "lumped species parameter" file, which is to be read by the airshed modeling program during the mechanism initialization process. The program prompts the user to give the name of this output file. (The user can optionally have the program output the composition of lumped model species corresponding to the representative emissions profile as a separate file from that containing the mechanistic parameters of the lumped classes, if that is desired. As indicated above, these representative composition data are not required by the airshed model program, but can be used as defaults when no composition data are required, if that is determined to be appropriate.)

(3) In the third and last step, the program uses the data in the SAROAD.PRM file and the lumping control file (input in the previous two steps) to produce the lumped SAROAD assignment file, which is used as input to the LUMPEMIT program to process the emissions input for the airshed modeling program. For each SAROAD class, the program uses the data in the SAROAD.PRM file to determine which (and how many moles) of the detailed model species are used to represent it, and then uses the data in the lumping control file to determine how these detailed model species are lumped as lumped model species. The program then outputs in the SAROAD assignment file which and how many moles of detailed model species are used for each kg of emissions of the SAROAD class. For some lumped classes, the amount of detailed model species associated with them may depend on the relative rate constants of each, and the rate constant for the lumped class may depend on the representative emissions profile input

for the first step of the program. However, for SAROAD classes represented only by explicit model species, or by lumped groups for which no rate constant weighing is done in the averaging (see Section III.D.2), the assignments in the SAROAD assignment file will not depend on the representative emissions profile. The program prompts the user for the name for the SAROAD assignment file to be output; if the user chooses not to give a name in response to this prompt, then this step is bypassed.

b. Running the Program

Although the PREPEMIT program operates logically in the order described above, the program actually prompts for and then reads the lumping control file before reading and processing the representative emissions profile. Thus, when the PREPEMIT program is first invoked, it will output the prompt

```
File name for control parms.(<CR> to end):
```

In response to this prompt, the user should give the name of a valid lumping control file. (The format and use of the data in this file is described in the following section.) The program will then read this file and inform the user of how many "substitutions" (name changes or species represented by other species) and how many detailed model species were read. The program will then read the SAROAD.PRM file and then inform the user of how many SAROAD classes were assigned detailed model species in the file. (SAROAD classes for which there are no detailed model species assignments are ignored.) If the lumping control file was such that INTOH input was required but no such input was supplied in the lumping control file, (see Sections III.C.3.a, III.C.3.d, and III.D.2), the program will output the prompt

```
Give integrated OH in ppt-min. (50 is typical, negative =
infinite.)
```

The user then should input the integrated OH value that normally would go in the INTOH input in the lumping control file. (However, recommended usage is that if INTOH input is required, it should be in the lumping control file, and this prompt would not appear.)

After the lumping control data is input, the program will then output the prompt

Emissions file name:

The user should then give the name of an emissions input file, whose data is in the format described in the following section (for emissions input to both the PREPEMIT and the LUMPEMIT programs.) Then it will output the prompt

Output file for lumped parameters:

The user then should give a file name to use for the output file which will give the mechanistic parameters for the lumped species. The program will then prompt

Output file for lumped species, if different:

If the user wishes the lumped species compositions of the representative emissions profile to be output in a separate data set from the lumped parameters, the name of a file for these data should be given. In this case, the lumped species composition of the representative profile will be output in the same format as it would if it were prepared by the LUMPEMIT program. Otherwise, just enter <CR> at this prompt, and these data will be output as "comments" in the file with the lumped parameters. The program will then output the prompt:

Normalize to 1 mole NMHC (y/n, default=no)?

This determines whether the lumped species composition data for the representative emissions profile are to be normalized to one mole carbon (not counting CO and methane, if these are in the emissions file). If the user wants these data to be normalized, he should enter "Y"; otherwise, the composition data are not normalized. This does not affect the mechanistic parameters or the SAROAD assignments, i.e., it does not affect any of the data sets read by the LUMPEMIT or the mechanism initialization routines for the airshed modeling program.

Finally, after the program has finished calculating the lumped species compositions and the mechanistic parameters for the lumped classes and output the data sets named above, it gives the prompt

Output file for lumped SAROAD assignments:

If a file with lumped SAROAD assignments, required by LUMPEMIT, is required, the name of this output file is entered at this point. If no name is input, the program terminates without producing such a file. If a file name is given, the program produces the lumped SAROAD assignment file, and then terminates.

2. LUMPEMIT Program

The LUMPEMIT program is a simplified representative of a pre-processing program for emissions input into airshed modeling programs, which is used to illustrate how such programs can be interfaced into the overall emissions processing system described in this document. It contains two example subroutines which can be incorporated into existing emissions pre-processing programs for the purpose of converting emissions input data (mass emissions of SAROAD classes) into processed emissions data (molar emissions of lumped model species) as required by the airshed modeling program. The first example subroutine is an initialization routine which is called once at the start of the execution of the program to read the lumped SAROAD assignment file (output by PREPEMIT), which tells the program how to process the emissions. The second subroutine is the conversion routine which is called for each separate set of emissions input into the model, and actually does the conversions from the emissions input data to the processed emissions output data, based on the data read by the initialization routine. Programming aspects for interfacing these subroutines into existing emissions pre-processing systems are discussed separately in Section IV.B.2. The operations of the example LUMPEMIT program included with this software, which illustrates the use and operations of these subroutines in the processing of emissions data, and can be used for testing the implementation of the PREPEMIT program and its various input files, is discussed in this section. Considerations involved in running this version of the LUMPEMIT program must also be taken into account when utilizing other versions of emissions pre-processing programs which utilize this emissions processing system.

When the LUMPEMIT program is first invoked, the program will first output the prompt

Conversion factor (<CR> = use 1.0):

At this prompt, the user can optionally give a conversion factor to use when processing the emissions data. All processed emissions data are multiplied by this factor. The factors given in the SAROAD assignment output file output by PREPEMIT are based on the assumption that the input emissions data are given in terms of kg per unit time for the SAROAD classes, and that the processed emissions data to be output are to be given in terms of moles per same time unit for the lumped model species. If some other units are to be used for the input and/or output emissions data, then the appropriate factor should be input at this point. (Note that if the output units are to be given in terms of PPM rather than moles, this conversion factor will depend on temperature and total pressure.) Alternatively, the user could (and probably should) modify the source code of the program so that it always employs the appropriate conversion factor corresponding to the input emissions data and desired processed emissions output.

The LUMPEMIT program will then output the prompt

SAROAD assignment file:

The user then must give the name of a SAROAD assignment file which has already been produced by the PREPEMIT program. The file named must have been produced by running PREPEMIT with the input data appropriate to the particular application of the airshed model for which the emissions are being processed. (I.e., the "representative" emissions profile input into PREPEMIT must be appropriate for the air basin to be modeled and the lumping control file must be compatible with the chemical mechanism in the airshed model). After this file name is input, the LUMPEMIT program will then call the initialization subroutine to read the data in this file to determine how the emissions are to be processed.

The program will then output the following prompt, asking about the format of the output data which is desired:

Output emissions amounts with species names in
separate files (No)?

If the user desires the processed emissions data to be in a output in a single data set for all the profiles input during the run, in a condensed form suitable for use with multi-celled airshed models, the default answer ("NO", or "N") should be selected. If this is selected, the program will then output the prompt

Output file name =

requesting the name of the file for the processed emissions data. The output data in this file will consist of sets of arrays of numbers, one array for each input profile, giving the compositions of lumped model species which correspond to the input profile, with the species ordered in the same way as they are in the mechanistic parameter files. The format of the processed emissions data sets produced if this option is selected is discussed in Section IV.B.3.c.

If the user desires the processed emissions data to be output in separate files, one for each emissions profile input, he should answer "Y" (or "YES") to the above prompt. This would result in the program prompting for the name of an output file for each input emissions file, as indicated below. The data in these separate files would consist first of comments (with a "!" in column 1 of each record to indicate it as being a comment) giving the name of the lumped SAROAD assignment file, and the emissions input file used to create it, and a summary of the total mass, and total mass of unknowns (SAROAD species for which there is no assignment in SAROAD.PRM, and thus there is no corresponding assignments in the lumped SARPAD assignment file), and then one record for each lumped model species, giving first the name of the species, then its emissions level. This format is useful for single-cell model simulations such as those discussed in Section V, and is compatible with the SAPRC model simulation program which was used to test the implemtation of this software. (Examples of processed emissions output files are the .EMO files discussed in Section V and included among the files distributed with this software listed in Appendix A). However, this is probably not a useful format for processed emissions data for input into multi-celled airshed models, because of the large number of emissions profiles which need to be input into such models.

After these intialization steps are completed, the program will then repeatedly ask the user for the name of an input emissions data file and then the name of an output file for the processed emissions data. For example, for the first set of emissions data, the program will first prompt

```
Emissions input set #  1:
```

asking for the name of an input emissions file. If the option to have separate output files for each emissions profile was selected, the program will then also give the prompt

```
Emissions output set #  1:
```

requesting for the name of the output file. (This prompt will not appear if they are all to be in the same file, since the program has already obtained the name of that file, as indicated above). After the necessary file name(s) are obtained, the program will then read the input file named (assumed to contain mass emissions for the SAROAD classes, in the format described in the following section for "emissions input data", and then convert these data to emissions in terms of molar [or ppm] emissions of lumped model species, and output these into the named file in the format described in the following section for "processed emissions data"). The program will continually loop back and ask for names of additional emissions input data to process (and names of output files to write the processed data, if needed) until the user gives a blank input file name. If the user gives a blank input file name, the program will terminate. If the user gives a blank output file name, the program will ask for the output file name again.

C. Formats of Input Data Sets

In this section, the formats of the input data sets used as input by the PREPEMIT and LUMPEMIT programs, and which must be created, updated, and/or modified by the user, are described. These include (1) the emissions input data sets; (2) the SAROAD.PRM data set; (3) the lumping control files; and (4) the mechanistic parameter files. There are several other types of data sets employed in the system of emissions processing programs, but these are created by the programs and are not intended to be updated or maintained directly by the user. The formats of these data sets are discussed in Section IV.B.3 in conjunction with the overall description of the software components.

1. Emissions Input Data Sets

As discussed above, the emissions input data sets simply contain the mass emissions of the various SAROAD classes which are used to categorize the raw emissions data. As used by the present software system, each emissions data set contains the data for only a single emissions profile. These data sets are used as input to both the PREPEMIT and the LUMPEMIT program, and in both cases the formats of the data are the same, though the use of the data by the two programs are different. In the case of input into the PREPEMIT program, the input emissions data set consists of the composition of an emissions profile which is taken as representative of the entire airshed modeling region. Typically, this would be the sum of all emissions into the modeling region. In the case of input into LUMPEMIT, the emissions input data are intended to represent only the emissions into a single region of the model at a given time.

Emissions input data sets can contain three types of input records: (1) comments, which are indicated by a "!" character in the first column, which are ignored by the programs; (2) conversion factor records; and (3) emissions data records. Input is terminated only by end-of-file, and blank input records are ignored. The emissions data record consists of a five-digit integer followed by a floating point number (FORTRAN code = I5,G11.2). The integer is the SAROAD ID number, and the floating point number is the mass emissions. The conversion factor record has the characters "FAC=" in the first four columns, followed by a conversion factor (in G11.0 format). All subsequent emissions input data are multiplied by this factor. If a second conversion factor record is input, the subsequent data are multiplied by both factors. If a conversion factor of zero is input, all input data are ignored until the next conversion factor record.

2. SAROAD.PRM Data Set

The SARAOD.PRM data set gives the detailed model species assignments and other information for the SAROAD emissions classes, including all the data needed by the PREPEMIT program to convert data given in terms of mass emissions of SAROAD classes into molar emissions of detailed model species. It also contains comments describing the assignments and special problems or considerations involved, and codes giving estimates of uncertainties associated with these assignments. The SAROAD.PRM data set was created as part of the development of this emissions processing system, and contains assignments for most of the SAROAD classes in the 1983 California emissions inventories for which there are non-zero emissions, as well as for many of the major SAROAD classes currently used by the EPA in the current NAPAP emissions inventory for the contiguous United States.

The SAROAD.PRM data set is an ASCII text file which consists of two parts. The first part consists only of comments, which are ignored by the PREPEMIT program. These consist of all records in the data set until a record with a "." character in the first column. In the present data set, these comments describe the format of the data in this data set, and the meanings of the code numbers used for notes and uncertainty classifications (see below) -- i.e., it has essentially the same information as given in this section.

All records following the first record with the "." in the first column are the data records. These consist of 1-3 groups of records, containing data as described in Table 4. That table also indicates the meanings for the various codes numbers used to give uncertainty estimates and notes related to the assignments made for the SAROAD classes.

3. Lumping Control File

The lumping control file is an ASCII text file which controls how the PREPEMIT program assigns lumped model species to the detailed model species, controls which species are lumped with which, controls the input of the mechanistic parameters for the detailed model species, and serves other major functions affecting how compositions of lumped model species, and their kinetic and mechanistic parameters, are calculated by PREPEMIT. These files can contain "comment" records (indicated by a "!" character in the first column) to document its contents, and a number of types of input records, or "commands," which are summarized in Table 5. These commands do not necessarily have to be in any particular order within the lumping control file, except when indicated to the contrary in the following discussion. These different types of input records, and considerations and concepts involved in their use, are discussed in the following sections.

a. GROUP Command

The GROUP and PARM commands are used to define the characteristics of the lumped groups, and to determine which detailed model species are represented by them. In the context of this software, a "lumped group" is a special type of lumped model species which represents, in the airshed model, a set of detailed model species which (a) are of the same general chemical type, identified by the "lumped group type ID number" associated with them in the GROUP and PARM commands (see below), and (b) which react with OH radicals within a similar range of rate constants. The kinetic and mechanistic parameters of the lumped groups are derived based on the averages of those lumped model species which they represent, weighed by the amounts present (or estimated to be reacting) in the representative emissions profile. The GROUP command is used to define the characteristics of the lumped groups in the mechanism, while the PARM command is used to associate a set of detailed model species to a particular set of lumped groups, and to inform the program where to obtain the kinetic and mechanistic parameters for those detailed model species. The GROUP command is summarized briefly in this section, and the PARM command is summarized in the following section.

It should be noted that the proper use of these commands requires a general understanding of the lumping approach employed with lumped groups, and the necessary correspondences between the parameters in these commands and the data in the mechanism preparation program input specifying the reactions of the lumped groups and the data in the mechanistic parameter files referenced by the PARM commands. These are discussed in Section III.D, and users should be familiar with the contents of that section before modifying GROUP or PARM commands in lumping control files.

Table 4. Format of Data in the SAROAD.PRM Data Set, Giving Detailed Model Species and Other Assignments for the SAROAD Emissions Classes

FORTRAN Format Code	Name (a)	Description
		First record for each SAROAD Class:
I6,2X	SAR *	SAROAD Code No.
A30	NAME	Name or description of SAROAD class
I3	NMIX *	Number of detailed model species assigned to this SAROAD class. (0 = No assignments made for this SAROAD class.)
I3	KOHCLS	Uncertainty code for OH radical rate constant (kOH) 0 = See uncertainty code for mechanism 1 = Recommended based on multiple measurements, or clearly negligible 2 = Measured 3 = Estimated using established methods 4 = Uncertain 5 = Highly uncertain
I3	MECCLS	Uncertainty code for the mechanism 0 = Uncertainty code not assigned 1 = Mechanism considered well established (e.g., simple alkanes, alkenes, acetaldehyde, MEK, methanol, etc.) 2 = Mechanism not as well established. (e.g., >= C6 alkanes, simple ethers, cycloalkanes, >= C6 alkenes, alkylbenzenes, naphthalene) 3 = Mechanism or appropriateness of representation uncertain (e.g., dienes such as isoprene, cycloalkenes, pinenes, alcohol ethers, esters, C4+ aldehydes) 4 = Mechanism or appropriateness or representation highly uncertain (e.g., acrolein, >= C8 aldehydes, Cl, Br, I- containing organics, "odd" aromatics) 5 = Mechanism highly uncertain or representation used probably inappropriate (e.g., dimethyl formamide, amines)
I3	NOTE	Footnote concerning representation of SAROAD class 0 = Not assigned 1 = Explicitly represented 2 = Lumped with other compounds believed to have similar reactivities and mechanisms.

(continued)

Table 4 (continued) - 2

FORTRAN Format Code	Name (a)	Description
		3 = Mixture -- typical composition given 4 = Composition of mixture is uncertain or ambiguous 5 = Extremely poorly characterized mixture.
		(The following included only if NMIX=0, i.e., if no model species assigned)
6X,I3	NC	Number of Carbons (or zero if number of carbons has not been assigned)
F8.2	MWT	Molecular Weight (or zero if molecular weight has not been assigned)
		(The following included only for unassigned SAROAD classes used by the EPA)
F12.2	EMIT	Mass emissions of this SAROAD class in the contiguous United States
	Next NMIX records for each SAROAD class	(No input if NMIX = 0)
6X	(blank)*	(Must be blank)
F7.4	LVL *	Weight fraction of model species for this SAROAD class
1X,A8	SPCLAS *	Species Mechanism Class (detailed model species)
1PE12.3	AOH *	Arrhenius A factor or rate constant for OH reaction (cm^3 $molecule^{-1}$ sec^{-1} units). If zero, default for the detailed model species is used.
0PF7.3	EA *	Activation energy (kcal/mole) for OH reaction. (Ignored if AOH is zero)
F6.3	B *	B-factor for OH reaction. (Ignored if AOH is zero) [kOH = A * ((T/300)**B) * exp(Ea/RT)]
1X,I2	NCI *	Number of carbons in this model species
F8.2	MWI *	Molecular weight of this model species
I3	ITI	Detailed chemical class code (b)

(continued)

Table 4 (continued) - 3

FORTRAN Format Code	Name (a)	Description
I3	ITI2	Detailed chemical subgroup code (b)
	Comment records for SAROAD class. (All following records with no data in col 1-6 are treated as comments and are ignored. These are optional, and are only included for SAROAD classes where additional documentation is included.)	
6X	(blank)	(Must be blank)
A74	COMT	Comment

[a] Data which are required by PREPEMIT are indicated by a "*" following the name. Other data are ignored by PREPEMIT.
[b] Detailed chemical class codes are used internally at SAPRC, and are not used by PREPEMIT.

Table 5. Summary of Commands and Input Records Which Can Be Included In Lumping Control Files (a)

Command	Description and Data Format
! text	Comments. (Ignored by program)
GROUP	Defines lumped groups, the species and parameter names used for them, and the minimum and maximum OH radical rate constants for species to be lumped together in the same group.

Col.	Data
7	Lumped group type ID number Used to identify sets of lumped groups where model species can be lumped together based on their OH radical rate constants. Can be any number between 1 and 9.
9-12	Four-character name for this lumped group in the mechanism.
14-15	Two-character parameter prefix for mechanistic parameters for this group
17	Emissions weighing code number. This indicates how much emissions levels of the detailed model species contribute to emissions of this lumped group. (0 = weighed by amount emitted, i.e., no rate constant weighing in calculating contribution of species to lumped group; 1 = weighed by amount reacted, estimated using OH radical rate constants and integrated OH input using the INTOH command; 2 = weighed by amount reacted, assuming amount reacted is proportional to the OH radical rate constant)
18-27	Minimum OH rate constant for model species to be lumped into this lumped group.
28-37	OH rate constant to use for this lumped group if rate constant weighing is used. Ignored if no rate constant weighing done. If zero or blank, then rate constant for group is calculated based on those for detailed model species in the representative emissions profile which are assigned to this group.

Command	Description and Data Format
PARM	Controls input of kinetic and mechanistic parameters for detailed model species to be lumped into the lumped classes

Col.	Data
6	Lumped group type ID number. Determines which sets of groups these model species are lumped with. Corresponds to lumped group type ID number on col. 7 of the "GROUP" input records.

(continued)

Table 5 (continued) - 2

Command	Description and Data Format
	8 — File format code: (1 = general, 2 = alkane, 3 = alkene)
	10-49 — Name of file containing detailed model species names and their kinetic and mechanistic parameters. Format for data in these files is described in Section III.C.4.

TEMP

Defines Temperature to use for rate constant calculations

Col.	Data
5-16	Temperature to use (K)

Note that the TEMP command must precede any PARM commands in the file. The default temperature is 300 K, and no TEMP command is required if that is used.

INTOH

Defines integrated OH radical concentration to use for calculating estimated amount of model species reacted in model simulation. Used when rate constant weighing option "1" is selected in the "GROUP" command.

Col.	Data
6-20	Integrated OH in ppt-min

REPLACE

Used to replace one detailed model species name with another. Useful when more than one name is used for a model species, or if a detailed model species is to be represented by an explicit model species in the mechanism (as opposed to being included in a lumped group). Commands for name replacements, or substitutions, are input from a named file, or from the lumping control file following this command.

Col.	Data
9-38	File name to read replacement or substitution commands. If blank, these commands are read from the subsequent records in the lumping control file unil an "END" record is input.

Format for name replacement commands (whether in the named file or the lumping control file) are as follows. Note that all records with a "!" in column 1 are treated as comments in either case.

(continued)

Table 5 (continued) - 3

Command	Description and Data Format

Col.	Data
1-8	Detailed model species name to be replaced
10-17	Detailed or lumped model species name to use for species named in col. 1-8.
18-27	Factor to multiply species concentration by when making replacement. If blank or zero, factor used is 1.0

SPECIES — Used to specify explicit model species in the mechanism, and input their molecular weights and number of carbons, so these data can be passed to the airshed modeling program. Model species specifications are in the named file, or will be in the lumping control file following this command.

Col.	Data
9-38	File name to read explicit model species specifications. If blank, these data are read from the subsequent records in the lumping control file unil an "END" record is input

Format of explicit model species specification commands is as follows. As with "REPLACE" input, the same format is used regardless of whether the data is taken from a named file or directly from the lumping control file, and comments, indicated by a "!" in column 1, are allowed.

Col.	Data
1-8	Explicit model species name. (Names up to 8 characters are permitted, but many airshed modeling programs require names to be 4 characters or less. Must be a species in the mechanism.)
9-10	Number of carbons
11-18	Molecular weight (grams mole^{-1})

END — Used to terminate "REPLACE" or "SPECIES" input if they are taken from the lumping control file.

(continued)

Table 5 (continued) - 4 -

Command	Description and Data Format
	(The following options are included for compatability with SAPRC modeling software only, and must not be included if the parameter files output by PREPEMIT are to be read into an airshed modeling program using the RDPRM interface subroutine which is described in this document. If any of these options are used, then lumped SAROAD assignment files will not be produced -- and the user will not be prompted to give their names. These options are summarized on this table for completeness, but their use is not documented here.)
INTFMT	Causes some of parameters output in the mechanistic parameter files to be output as comments (i.e., with a "!" character at the beginning of the record. The affected output are summaries of molecular weights and numbers of carbons of the model species, and levels of "NROG". This permits these parameter output files to be used as input to the SAPRC model integration program, which does not recognize the affected data as legal input.
PREFIX	Causes a four-character (maximuum) prefix to be placed before the names of detailed model species when their emissions levels are output. Used for certain applications at SAPRC. Does not affect symbols used for mechanistic parameter output.

Col.	Data
8-11	Prefix character(s). First character must be non-blank.

As indicated above, the GROUP command is used to define the major characteristics of the lumped groups, with one GROUP command for each lumped group in the mechanism. The parameters in this command are used to (1) identify the general types of chemicals which can be included in the lumped group; (2) give the symbol used for the lumped group in the mechanism; (3) give the prefix used for the symbols used in the mechanism for the mechanistic parameters of the lumped group; (4) specify the type of emissions weighing approach to employ in calculating contributions of detailed model species to this lumped group; (5) specify the range of OH radical rate constants for species which can be represented in the lumped group; and (6) optionally give the OH radical rate constant for the lumped group, if it is not to be calculated based on the representative emissions profile.

The first parameter in the GROUP command is a single digit code number (1-9) which is designated the "lumped group type ID number." This number is used to identify a set of chemicals (or detailed model species) which can be represented by this lumped group, provided they also react with OH within the range of rate constants associated with the lumped group, as indicated below. This number is associated with a set of detailed model species using the PARM command, as indicated in the following section. Thus, for example, if a GROUP command has a lumped group type ID code of "5," only the detailed model species included in mechanistic parameter data sets referenced by PARM commands with a group type ID code of "5" can be represented by the lumped group defined by this GROUP command. In general, there can be more than one GROUP command with the same group type ID number, with each of these representing species with a different range of OH radical rate constants (see below and Section III.D).

The second parameter is a four-character alphanumeric code which is the name used in the mechanism for the lumped group. It must be unique (i.e., not the same as any other lumped group or explicit model species in the mechanism), and the mechanism employed must include reactions for this species in the proper format for lumped groups, and define all the mechanistic parameters employed for the set of detailed model species it is used to represent. (See Section III.D.3.)

The third parameter is a two-character alphanumeric code which is the prefix used for the codes for all the mechanistic parameters in this lumped group. As indicated in Section III.D.3, all mechanistic parameters for lumped groups have four-character codes, with the first two identifying the lumped group, and the second two identifying the type of parameter, as they are referenced in the mechanistic parameter files. This must not be the same as the parameter code for any other lumped group, and must correspond to the codes used in the mechanism for the parameters for this lumped group.

The fourth parameter is a single digit code ("0", "1", or "2") which indicates how contributions of emissions of individual species represented by this lumped group are to be calculated. Briefly, "0" is used if the emissions are to be weighed by the amount emitted without any consideration of how rapidly they react, "1" and "2" are used if the emissions are to be weighed by the estimated amount reacted, with OH radicals, in the airshed simulation. If "1" is used, the estimate is based on the integrated OH radical levels in the model simulation, which must be input using the "INTOH" command in the lumping control file, while "2" is used if it is assumed that all species represented by this lumped group react sufficiently slowly that the amount reacted can be assumed to be simply proportional to their OH radical rate constant. See Section III.D.2 for a more detailed discussion of this.

The fourth parameter is the minimum OH radical rate constant for the reactions the model species to be represented by this lumped group. It must be in units of ppm^{-1} min^{-1} since these are assumed to be the units employed in the airshed model. If no minimum OH rate constant is given in the GROUP command, then the effective minimum OH rate constant is zero -- i.e, the lumped group is for the slowest reacting species associated with the group type ID of the group. The maximum OH rate constant for species which are lumped into the group is the lowest minimum OH rate constant given in the other GROUP commands with the same type ID number which is greater than the minimum specified for this group. If there is no other lumped group with the same type ID with a higher minimum OH rate constant, then this lumped group is used for the fastest reacting species of this type. Obviously, no two lumped groups with the same type ID should have the same minimum OH rate constant specified in the GROUP command, otherwise one of those would never have any species lumped into it. It is also strongly recommended that all GROUP commands for the same type ID be given in the lumping control file in order of ascending minimum OH radical rate constant, since it is not guaranteed that erroneous results may not occur if GROUP commands of the same type are not ordered in this way. However, the program will not detect an error if this ordering is not followed, and the results may not necessarily be erroneous.

The final parameter is the OH radical rate constant to use for the lumped group, which can be used only if weighing of emissions of species represented by this lumped group are based on estimate amounts reacted, i.e., if the fourth parameter codes are "1" or "2." If no rate constant is input here (or if a rate constant of 0 is input), then the rate constant for the lumped group is calculated based on the distribution of detailed model species lumped into this group in the representative emissions profile input into PREPEMIT. If no rate constant weighing of emissions into this group is done (fourth parameter code is "0"), the rate constant for the lumped group is always calculated based on emissions in the representative profile, and this parameter is ignored.

b. <u>PARM Command</u>

The PARM command is used to associate a set of detailed model species with a particular set of lumped groups, and to inform the PREPEMIT program where and how to read the list of detailed model species and their kinetic and mechanistic parameters. Names and parameters for all detailed model species which are to be represented by lumped groups must be specified by data sets referenced by a PARM command. (However, detailed model species which are represented by explicit species in the mechanism are not included in these data sets, but are referenced using the REPLACE or SPECIES command, as discussed below.) This command has three parameters, as discussed below.

The first parameter is the single digit (1-9) lumped group type ID number, which is to be associated with all the detailed model species listed in the mechanistic parameter data set. As discussed in the previous section, this is used to control which lumped groups can be used to identify these detailed model species. There is no requirement as to which of these ID numbers are used and their ordering, except that any group ID numbers used in a PARM command must also be used on at least one

GROUP command. Otherwise, the PREPEMIT program would not be able to associate these detailed model species with any lumped group. Likewise, for any group ID number used in a GROUP command, the same number must appear on a PARM command, or there would be no detailed model species for the lumped group. More than one PARM command can have the same group type ID number, so detailed model species which are lumped together do not all have to be included in the same mechanistic parameter data set.

The second parameter in the PARM command is a single digit code number (currently "1", "2", or "3") giving the format of the data in the mechanistic parameter data set referenced by this command. The legal data formats for mechanistic parameter data sets are discussed in Section III.C.4. The PREPEMIT program will abort if a format code other than "1," "2," or "3" is given. Format codes "1" and "2" are appropriate for alkanes, aromatics, and other species which react primarily with OH radicals, while format code "3" is appropriate only for alkenes. Because alkenes have different types of kinetic and mechanistic parameters than species which react primarily with OH radicals, and detailed model species input using format "3" should not be lumped with species input using the other two formats.

The final parameter in the PARM command is the full name of the data set from which the detailed model species and their parameters are to be read. The PREPEMIT program will abort if it cannot find or read this data set, or the data in it are not in the format consistent with the format code given as the second parameter.

c. TEMP Command

In general, rate constants and mechanistic parameters for the detailed model species can depend on temperature, and the temperature dependences for these parameters can optionally be included in the mechanistic parameter data files. However, the present version of the PREPEMIT program does not allow for temperature-varying parameters, and all rate constants and parameters for lumped groups must be calculated for a given temperature. The TEMP command is used to tell PREPEMIT what temperature to use to calculate values for parameters for which temperature-dependence data are given. If no TEMP command is included in the lumping control file, the default temperature of 300 K (26.8°C) is used.

Note that when the data from the mechanistic parameter files are being processed by LUMPCOMP (in response to the PARM input in the lumping control file), the present version of the program calculates the values of the temperature-dependent parameters immediately, and does not save the information on the temperature dependence for these parameters. This means that if a temperature other than the default of 300 K is to be used, the TEMP input must be given before any PARM input. If a TEMP command is given after a PARM command, the program has no way of changing the values of temperature dependent parameters which were already input to the values appropriate for the new temperature, and thus the program will abort with an appropriate error message. This is one of the few cases where there is a restriction on ordering of input commands in lumping control files. The recommended practice is for the TEMP command to be one of the first commands input in a lumping control file.

d. INTOH Command

As indicated above in the discussion of the GROUP command, and in more detail in Section III.D.2, emissions into lumped groups can optionally be weighed by the estimated fraction of the emitted species which reacts under the conditions of the airshed simulation, which in turn can optionally be estimated from its OH radical rate constant and the integrated OH radical levels characteristic of the model simulation. The INTOH command is used to inform the PREPEMIT program of the integrated OH radical levels to be used in calculating the contributions of emissions for species associated with lumped groups for which this option is specified. The units of INTOH which are expected by the program are ppt-min. (A typical value for a 1-day simple box model simulation representative of an ozone pollution episode is 50 ppt-min, which corresponds to a 12-hour average OH radical level of 7×10^{-8} ppm, or 2×10^{6} radicals cm^{-3}.) The way this quantity is used by this program is described in Section III.D.2.

Note that INTOH input is required by the program only if some lumped group has an emissions weighing code of "1" specified in its GROUP command. If no such option is specified, the INTOH input is ignored. If this weighing code is specified for some lumped group, but there is no INTOH input in the lumping control file, the program will prompt the user to supply an INTOH value. (If a negative number is given for the integrated OH level, then the effect is the same as changing the weighing codes of "1" to weighing codes of "0", i.e., no rate constant weighing is used for those lumped groups.) This prompt will appear following the prompt for the lumping control file.

e. SPECIES Command

The SPECIES command is used to list all the explicit model species in the mechanism, and their numbers of carbons and molecular weights. By "explicit model species," we mean all species in the lumped mechanism which can be emitted except for the lumped groups. (They are called "explicit" because their parameters are not adjusted based on the group of emitted species they are representing -- the reactions and mechanistic parameters for them are given explicitly in the mechanism.) The PREPEMIT program includes these data in the mechanistic parameter output file which is read by the airshed modeling program, making these data available to that program. In addition, the program requires the molecular weights for the explicit model species so it can output the factors required to calculate molar emissions for such species from mass emissions of the SAROAD classes in the file giving the lumped model species assignments for the SAROAD classes. (The number of carbons and molecular weights for the lumped groups are not specified because these are calculated based on the set of detailed model species in the representative profile which they represent.)

The SPECIES command can optionally include as a parameter the name of a file from which the species specification input will be read. If no file name is given, the PREPEMIT program will interpret all subsequent input from the lumping control file as species specification commands, until an "END" command is input. After the "END" command is input, the PREPEMIT program will then expect normal lumping control commands. It is also legal to terminate input from the lumping control file by the end-of-file of the lumping control file, if no further lumping control commands are required. However, if a lumping control command is encountered following a SPECIES command without a preceding "END" command, an error will occur (or, worse, the program may erroneously interpret the lumping control command as an explicit species specification input record.) If the species specification input is taken from a file, the format of the data is the same as if it were taken from the lumping control file, except that "END" commands and other lumping control commands are not permitted; the end-of-file is the only way the input is terminated if the input is from a separate file. In either case, comments (records starting with a

"!" character) are permissible when species specification input is expected, as they are anywhere else in lumping control files.

Species input records have three parameters. The first consists of 1-8 alphanumeric characters which is the name of the explicit model species in the mechanism. The mechanism must include reactions for species with this name, with all the mechanistic parameters (if any) defined with their appropriate values for the mechanism. It can be the same as the name as a detailed model species used in the SAROAD assignment file, though this is not necessary, since one or more detailed model species can be associated with one or more explicit model species using the REPLACE command, discussed below. It must not be the name of a lumped group. The second and third parameters in the species specification input is the number of carbons and the molecular weight, respectively, for the explicit model species. The number of carbons must be an integer and can be zero. The molecular weight should be given in units of grams per mole. Molecular weights of zero are not permitted.

The explicit model species name input in species specification input can be the same as the name of a detailed model species included in a mechanistic parameter file referenced by a PARM command. However, if this is the case, the specification of this species as an explicit model species in the mechanism overrides its specification as a detailed model species which can be lumped in a lumped group. Thus, detailed model species named in SPECIES commands are assumed to be explicitly represented in the model, and not lumped with any lumped group, regardless of whether or not they are included in a parameter file referenced by a PARM command. (If species specified as explicit are included in a mechanistic parameter file, their mechanistic parameter data in the file is ignored.) This result is the same regardless of the relative ordering of the SPECIES and the PARM command in the lumping control file.

f. REPLACE Command

The REPLACE command is used when one detailed model species is to be represented by another species. The species it is replaced by should be either another detailed model species included in a mechanistic parameter file referenced by a PARM command, an explicit model species referenced in the input following a SPECIES command, or a species which is replaced by such a species using other REPLACE input. The primary utility of the REPLACE command is when one or more detailed model species are to be represented in the mechanism by an explicit model species, using the "lumped molecule" approach. (The "lumped molecule" approach is the lumping approach where a compound is represented in the model by a chemically similar compound, without any adjustment of parameters for the compound in the model based on the particular species it is representing.) However, the REPLACE command is also useful when there is an inconsistency in the nomenclature of detailed model species in the SAROAD.PRM and the mechanistic parameter files, or when the mechanistic parameters for a number of detailed model species are the same. This command can be used not only for simple substitutions, but also when one molecule of a model species is to be represented by a different number of molecules of the other.

Like the SPECIES command, the REPLACE command can optionally include as a parameter the name of a file from which the species specification input will be read. If no file name is given, the PREPEMIT program will interpret all subsequent input from the lumping control file as REPLACE until an "END" command is input, or until the end-of-file. "END" commands are not permitted in replace input if they are from a file other than the lumping control file, but comments (input records beginning with a "!") are. The format of REPLACE input is the same regardless of whether they are from the lumping control file or a separate file named on the command.

The input records following a REPLACE command (other than comments) must have a minimum of two parameters and optionally have a third. The first parameter is the species name which is being replaced, and the second is the one it is being replaced by. Both must consist of 1-8 alphanumeric characters. The third parameter is optional, and is the number of moles of the second species which is used to replace one mole of the first. For example, the REPLACE input

```
ACETONE  MEK      0.5
```

causes each mole of the species "ACETONE" to be replaced by 0.5 moles of the species "MEK." If no third parameter is given, it is assumed to be one, e.g., if the "0.5" were omitted from the above input, each mole of "ACETONE" would be replaced by a mole of "MEK."

Note that when PREPEMIT processes detailed model species composition data, it processes all the REPLACE commands before any other processing, such as searching through lists of species names read from the mechanistic parameter data sets, is carried out. In particular, name substitutions caused by REPLACE commands are made before any specifications for detailed model species are read from parameter files referenced by PARM commands, regardless of the relative positions of the REPLACE and PARM commands in the lumping control file.

4. Mechanistic Parameter Files

Mechanistic parameter files are input files to the PREPEMIT program which give the names and mechanistic parameters for the detailed model species which are to be represented by the lumped groups in the mechanism. All species in a given mechanistic parameter file must be of the same chemical type, i.e., be species which can be represented by the same set of lumped groups based only on how rapidly they react with OH radicals (see Sections III.C.3.a-b, and III.D.1). By "mechanistic parameters," we mean (a) the OH radical rate constant, which must be specied for all species represented in lumped groups; (b) stoichiometric coefficients for products which can be formed when the species reacts; and, for alkenes only, (c) the rate constants for the reactions of the species with O_3, $O(^3P)$ atoms, and NO_3 radicals. Note that changes to the mechanistic parameter data files is essentially the same as changing the chemical mechanism used in the airshed model for the emitted species, and should be made only with due consideration of the atmospheric chemistry of the species involved, or the chemical validity of the model predictions may be affected.

As indicated in Table 5 and in Section III.C.4, there are presently three legal formats for mechanistic parameter files, with the format being indicated as a parameter in the PARM command where the file is referenced. For species which are represented in the mechanism as reacting only with OH radicals, either the "general" or the "alkane" format for the mechanistic parameter file can be used. In both cases, the only mechanistic parameters are the OH radical rate constant and the stoichiometric product yield coefficients. The latter are designated by two-character codes which are used to identify the product or products they represent. These codes must be consistent for all species which can be lumped together, and must be consistent with the codes used in the reaction mechanism for the lumped groups, as discussed in Section III.D.3.

The simplest of the two formats for mechanistic parameter files for species which react only with OH radicals in the mechanism is the "general" format, which is indicated by code number "1." This format can be used for inputs for all types of detailed model species whose stoichiometric product yield parameters are assumed to be temperature independent. This format is most appropriate for aromatics (where presently there are no data indicating temperature dependence of product yields), and can be used for alkanes and other saturated species if the product yields given in the files are for a temperature representative of the range of temperatures used in the model simulation. On the other hand, the "alkane" format, which is indicated by code number "2" can be used with model species whose stoichiometric product yield parameters may depend on temperature. This permits the same mechanistic parameter file to be used to derive parameters for lumped groups for a range of temperatures. Note, however, that the present version of this software does not permit information concerning the temperature dependence of the parameters of lumped groups to be passed to the airshed modeling program, and regardless of which format is employed, the parameters output would only be for the temperature specified in the TEMP input in the lumping control file. The data formats for the "general" and "alkane" mechanistic parameter files are given in Tables 6 and 7, respectively.

The "general" and "alkane" formats are not appropriate for alkenes because these species are represented in the mechanism as reacting with ozone, $O(^3P)$ atoms and NO_3 radicals as well as with OH radicals. For alkenes, the "alkene" format, indicated by code number of "3," must be used. The data formats for the "alkene" mechanistic parameter input files are given in Table 8.

Note that since the "general" and "alkane" formats are both used for inputting the same general types of mechanistic parameters, specifically the OH radical rate constant and stoichiometric product yield parameters, species whose parameters are input using either of these formats can be lumped together and represented by the same lumped group. (In other

Table 6. Organization of Mechanistic Parameter Data Sets which Use the "General" Format

FORTRAN Format Code	Data
	Group 1: 1 Record
80X	Comments identifying version of mechanism (ignored by PREPEMIT)
	Group 2: 1 Record
I2	Number of variable stoichiometric product coefficients
25(1X,A2)	Two character code identifying each of the stoichiometric product coefficients
	Groups 3: One set of three subgroups of records for each detailed model species. Terminated by end-of-file
	Subgroup 3a: First Record of set for each species:
A8	Detailed model species name
8X,I3	Number of carbons (ignored by PREPEMIT)
2I3	(Reserved for codes which are ignored by current version of PREPEMIT)
T31,3F10.0	Parameters A, Ea, and B for determining OH radical rate constant for detailed model species, where $k(OH) = A \times (T/300)^B \times e^{-Ea/RT}$ and T is the temperature in K (input using the TEMP command in the lumping control file), and R is 0.0019872 kcal deg^{-1} $mole^{-1}$. Units of A should be cm^3 $molecule^{-1}$ sec^{-1}, and units of Ea should be kcal $mole^{-1}$.
	Subgroup 3b: One record for each stoichiometric product yield parameter with a non-zero value. (Parameters with zero values need not be specified here, though such input is not illegal.) Such input continues until a blank record (Subgroup 3c) is input.
A2	2-Character code for mechanistic parameter. Must be one of those listed at the beginning of the file (Group 2).
F13.0	Mechanistic parameter value.

(continued)

Table 6 (continued) - 2

FORTRAN Format Code	Data
	Subgroup 3c: Input of parameters for a detailed model species is terminated by a blank record or end-of-file.
(blank)	(At least first two characters must be blank. Rest of record is ignored.

Table 7. Organization of Mechanistic Parameter Data Sets which Use the "Alkane" Format

FORTRAN Format Code	Data
	Group 1: 1 Record
5X	(Reserved for code identifying version of mechanism. Ignored by PREPEMIT
I1	Number of temperatures (NTEMPS). Must be 1, 2, or 3.
F9.2,2F10.2	Temperatures in K. Only first NTEMPS of these are used. Temperatures must be given in ascending order.
	Group 2: 1 Record
I2	Number of variable stoichiometric product coefficients (NPARM)
25(1X,A2)	Two character code identifying each of the NPARM stoichiometric product coefficients
	Group 3: 1 Set of NTEMPS+1 subgroups of records for each detailed model species. Group 3 input continues until end-of-file.
	Subgroup 3a: First Record of set for each species:
A8	Detailed model species name
8X,I3	Number of carbons (ignored by PREPEMIT).
I3	(Reserved for code not used by present version of PREPEMIT)
T31,3F10.0	OH Radical rate constant (in units of cm^3 $molecule^{-1}$ sec^{-1}) for each of the NTEMPS temperatures.
	Subgroups 3b: One set of records for each of the NTEMPS temperatures. INT(((NPARM-1)/8)+1) records for each temperature
(8F10.5)	Values of the NPARM mechanistic parameters, given in the same order as the parameter codes are given in Group 2, appropriate for the corresponding temperature.

Table 8. Organization of Mechanistic Parameter Data Sets which Use the "Alkene" Format

FORTRAN Format Code	Data
	Groups of one or five records for each alkene detailed model species.
	Record 1:
A8	Detailed model species name
8X,I3	Number of carbons (ignored by PREPEMIT).
2I3	Codes for substituents on each side of the double bond. Substituent codes are as follows: 0) (Used for cycloalkenes, as indicated below) 1) =CH2 2) =CHCH3 3) =CHR 4) =C(CH3)2 5) =C(CH3)R or =CR2 Where R is any substituent other than H or CH3. (If this is a cycloalkene, "0" should be used as the code for one of the substituents, and the other code should be either "3" if at least one side of the double bond is =CH-, and "5" otherwise.)
I3	Code to determine use of next parameter input, and to determine if kinetic parameters for this species are different from those input previously (IND).
G12.0	If IND is not negative, then this is the organic nitrate yield in the OH radical reaction. If IND is negative, then this is the glycolaldehyde yield in the OH radical reaction. (Negative values of IND are used only if the detailed model species is ethene.) The following records are input only if IND is not positive. If IND is positive, then this detailed model species is assumed to react with the same OH radical, ozone, NO_3 radical, and $O(^3P)$ atom rate constants as that input previously. (Note that a positive value of IND cannot be used for the first model species in the file.)
	Record 2 (input only if IND not positive):
G12.0,F8.0	A and Ea to use in calculating OH radical rate constant, where k(OH) = A x exp (-Ea/RT), and T is the temperature

(continued)

Table 8 (continued) - 2

FORTRAN Format Code	Data
	(input using the TEMP command in the lumping control file), R = 0.0019872 kcal deg^{-1} $mole^{-1}$, A is in cm^3 $molecule^{-1}$ sec^{-1}, and Ea is in kcal $mole^{-1}$ units.
60X	(Can be used for comments documenting source of rate constant)
	Record 3 (input only if IND not positive):
G12.0,F8.0	A and Ea for use in calculating the ozone rate constant, where the rate constant is calculated as shown above for the OH radical reaction.
60X	(Can be used for comments documenting source of rate constant)
	Record 4 (input only if IND not positive):
G12.0,F8.0	A and Ea for use in calculating the NO_3 radical rate constant, where the rate constant is calculated as shown above for the OH radical reaction.
60X	(Can be used for comments documenting source of rate constant)
	Record 5 (input only if IND not positive):
G12.0,F8.0	A and Ea for use in calculating the $O(^3P)$ atom rate constant, where the rate constant is calculated as shown above for the OH radical reaction.
60X	(Can be used for comments documenting source of rate constant)

words, PARM files referencing parameter files of each of these types can have the same group type ID number.) On the other hand, different types of mechanistic parameters are used for species in "alkene" format parameter files, and thus such species should not be lumped with species whose parameters are input using files with the other two formats. The various types of mechanistic parameters are discussed in Section III.D.3.

Comment records (records with a "!" as the first character, which are ignored by the program) are premitted in "general" and "alkene" format mechanistic parameter files. However, comment records are not permitted in "alkane" format parameter files.

D. Special Considerations Involving Lumped Groups

As indicated above, "Lumped Groups" are special types of lumped model species whose kinetic and mechanistic parameters are derived, by the PREPEMIT program, based on the set of detailed model species they are used to represent, and their relative emissions levels in the representative emissions profile. The PREPEMIT program allows the user a relatively high degree of flexibility in determining which sets of detailed model species are represented by the various lumped groups, and in determining how emissions of the individual species are weighed when PREPEMIT calculates the kinetic and mechanistic parameters for the lumped groups. This section gives a discussion of the options presented to the user in this regard, and some of the factors which should be considered in choosing among these options for various applications of the airshed model. In addition, in order for the output of the PREPEMIT and LUMPEMIT emissions processing programs to be successfully utilized by the airshed modeling program, there must be correspondences between the input data in the lumping control and mechanistic parameter files input by PREPEMIT and the input into the preparation program used to define the reactions of the lumped groups in the chemical mechanism used in the airshed model. These are also discussed in this section.

1. Determination of Which Detailed Model Species are Lumped Together

As discussed above in Sections III.C.3.a-b, the GROUP and PARM commands are used together to associate various sets of detailed model species to each of the lumped groups. Two factors are employed by the program to determine this: the group type ID number, and the OH radical rate constant. These factors, and considerations involved in their use, are discussed in this section.

The group type ID number is used to control the general types of chemicals which can be lumped together if they react at similar rates. In principle, any set of species which react only with OH radicals can be lumped together, provided that appropriate values for their different stoichiometric product parameters are also given. This means that alkanes, aromatic hydrocarbons, and many other types of saturated organics such as alcohols, ethers, esters, etc., which react in the atmosphere mainly with the OH radical, could be lumped together. On the other hand, these could not be lumped with the alkenes, since the alkenes also react to a non-negligible extent with ozone and (under some conditions) NO_3 radicals and $O(^3P)$ atoms. However, in practice, species with significantly different mechanistic reactivity characteristics, such as alkanes and aromatics (where the former tend to form intermediates which react to inhibit radicals, while the latter form highly reactive, radical initiating products), should only be lumped together if they are always emitted together in the same relative amounts at the various times and locations in the modeling region. Thus, for most airshed model applications, where emissions compositions can vary significantly from location to location and from time to time, it would not be advisable to lump together alkanes and aromatics. In such cases, the parameters for the alkanes and the aromatics should be included in separate mechanistic parameter files, and assigned different group type ID numbers in the PARM commands referencing those files, so they will be represented in the model by different sets of lumped groups. On the other hand, species such as alcohols, ethers, esters, etc., are estimated to have similar mechanistic reactivity characteristics as alkanes, and it may not be inappropriate to lump them in the same groups. This can be done either including such species in the same mechanistic parameter file as the alkanes, or using the same group type ID number in the PARM command referencing the files giving the mechanistic parameters for these species as used in the PARM command(s) referencing the file(s) containing the parameters for the alkanes.

For some airshed applications, it may also be appropriate not to lump biogenically emitted alkenes with those emitted anthropogenically. In particular, in applications where biogenics dominate the alkene emissions in one region of the modeling domain, while non-biogenics dominate in others, it is generally best that they be treated separately in the model. Otherwise, inappropriate mechanistic parameters for the affected alkene lumped group would be employed in both types of regions. On the other hand, if biogenic emissions are negligible throughout the entire modeling domain (or if they always dominate), or if the ratio of biogenic to anthropogenic emissions does not vary significantly, then it is not inappropriate to lump them together.

It should be emphasized at this point that in general the mechanistic parameters for a lumped group should not vary throughout the modeling domain. It is not valid for example, for a lumped group to have biogenic alkene parameters in one region and non-biogenic parameters in another in the same modeling run. If such were the case, then the transport of the lumped alkene from a region of biogenic emissions to a region of non-biogenic emissions would in effect result in the artificial conversion of the lumped species from a biogenic to a non-biogenic compound. It may not be inappropriate to have variable parameters for lumped groups in cases where the lumped group reacts so rapidly that transport from cell to cell is negligible. However, the emissions processing software and airshed program interface modules discussed here do not have the capability for lumped groups having different parameters in different regions of the model, and it is recommended that the programs not be modified to allow for this capability without due consideration of effects of transport of lumped species from one region of the modeling domain to another.

In general, there will be more than one lumped group associated with a given type of chemical (i.e., with a given group type ID number), with each being used to represent detailed model species which react with a given range of OH radical rate constants. There should be a sufficient number of groups for each type of chemical so that species with significantly different rates of reaction in the atmosphere are not lumped together. With respect to reaction rates in a given modeling scenario, species can be considered to fall into one of three classes, each with separate considerations with regard to how they are lumped: (a) slowly reacting species where only a relatively small fraction reacts during the model simulation; (b) rapidly reacting species where they are essentially completely reacted during a one-day simulation; and (c) species with intermediate reaction rates which fall in neither of the other two categories. In general, there should be a separate lumped group for each of these categories (provided, of course, that there are significant emissions into those categories). For the slower reacting species (class "a"), the reactivities are essentially proportional to their rate constant (all else being equal), and thus species with rate constants different than a factor of ~2 should not be lumped together unless rate constant weighing of emissions is employed. (Rate constant weighing is discussed in the following section.) On the other hand, only one lumped group should in most cases be necessary to represent the reactions of species in the fastest reacting class "c," since beyond a certain rate of reaction, the reactivity becomes independent of the reaction rate. This is because all of such species would react in the scenario in any case. Thus, rate constant weighing is not necessary for rapidly reacting species. For species of the intermediate type, the best approach may be to represent them in lumped groups where their emissions are weighed based on their rate constants and the estimated integrated OH radical levels for the modeling scenario (see below). However, the optimum lumping approach for species of intermediate reaction rates in complex airshed modeling simulations has not been investigated fully in this program, and it is recommended that alternative lumping schemes be investigated for conditions of representative airshed modeling conditions before settling on a lumping approach for routine use. The flexibility of lumped group definitions permitted by this software should allow this to be investigated relatively easily.

2. <u>Rate Constant Weighing of Emissions</u>

The PREPEMIT program permits three options as to how emissions of detailed model species are weighed in determining their contributions to emissions of the lumped group representing them. This is determined by the "emissions weighing code" parameter in the GROUP command defining the lumped group, as indicated in Table 5. Weighing method "0" is used when the contributions of emitted species to the lumped groups are simply weighed by the amount emitted. This is most appropriate for faster reacting species, where the amount reacted is essentially the same as the amount emitted. For slower reacting species, the amount reacted, and thus the contributions of the species to overall reactivity, will in general depend on how rapidly it reacts in the airshed model simulation. To take effects of reaction rate into account, either a sufficiently large number of lumped groups should be employed such that species reacting with significantly different rates are not lumped together, or the contributions of the emissions of species into their lumped groups should be weighed by the estimated extent to which the species react. Emissions weighing options "1" or "2" allow for weighing emissions of detailed model species by estimated amounts that they react in the model simulation, based on the assumption that they react primarily with OH radicals.

In general, if a species reacts primarily with OH radicals, the amount reacted in an airshed model simulation will depend on both its OH radical rate constant and the integrated OH radical levels in the simulation. Weighing method "1" can be employed to take both these factors into account. However, for lumped groups containing only very slowly reacting species, the relative amount of each of the species in the group which reacts will be simply proportional to their OH radical rate constant, and weighing method "2," which does not require estimates of integrated OH radical levels, can be employed. The specific manner in which emissions are weighed when PREPEMIT calculates kinetic and mechanistic parameters for the lumped groups, and when PREPEMIT and LUMPEMIT calculates contributions of detailed species emissions to emissions of their lumped groups, are discussed below for each of the three weighing methods.

If the emissions weighing code no. is "0," then the contributions of the emitted detailed species to their lumped groups are simply weighed by the amount emitted. This is the simplest method, since the weighing is independent of the nature of the species. If C_i is the molar emissions of a detailed model species i in a given lumped group, k_i is species i's OH radical rate constant, p_{ji} is species i's j'th mechanistic parameter, then the total emissions of (C_{tot}), the OH radical rate constant (k_{tot}), and the j'th mechanistic parameter (p_j) for the lumped group are given by

$$C_{tot} = \sum_i C_i$$

$$k_{tot} = \frac{\sum_i C_i k_i}{C_{tot}}$$

and

$$p_j = \frac{\sum_i C_i \, p_{ji}}{C_{tot}}$$

where the summations are over all the detailed model species (i) associated with the lumped group.

If emissions weighing methods "1" or "2" is chosen, then the contributions of detailed model species to their lumped group are weighed by the estimated amount of the detailed model species reacted during an airshed simulation, based on the assumption that they react only with OH radicals. If method "1" is chosen, then the amount reacted is estimated based on (1) the OH radical rate constant for the model species, and (2) INTOH, the estimated integrated OH radical concentration in the airshed model simulation (which is input by the user using the INTOH input command in the lumping control file as indicated in Table 5). The amount reacted is estimated using the approximation

$$\text{Model species (i) reacted} = C_i \left(1 - e^{-k_i \, \text{INTOH}} \right)$$

where (following the notation used above) C_i is the amount of species i emitted, k_i is its OH radical rate constant. Note that the above approximation is only strictly valid for simple scenarios where all emissions are present initially. This is obviously not the case in most complex airshed modeling applications. Nevertheless, for slower reacting species it can be a better approximation than assuming amounts reacted are proportional to amounts emitted, as is implied when method "0" is used.

If the above estimate of species (i) reacted is assumed, and if (using the notation employed above) C_i, k_i, and p_{ji} are the molar emissions, OH rate constant, and j'th mechanistic parameter for lumped model species i, and C_{tot}, k_{tot}, and p_j are the respective quantities for the lumped group, then C_{tot} and p_j are given by

$$C_{tot} = \frac{\sum_i C_i \left(1 - e^{- k_i \, INTOH} \right)}{\left(1 - e^{- k_{tot} \, INTOH} \right)}$$

and

$$p_j = \frac{\sum_i p_{ji} \, C_i \left(1 - e^{- k_i \, INTOH} \right)}{\sum_i C_i \left(1 - e^{- k_i \, INTOH} \right)}$$

where again the summations are over all detailed model species (i) lumped in this lumped group. In this case, the user has two options with regard to k_{tot}, the OH radical rate constant for the lumped group: (1) it can be specified explicitly as the fifth parameter in the GROUP command (as indicated in Table 5), or (2) if it is not given in the GROUP command, it is calculated, based on the emissions lumped in this group in the representative profile in a manner analogous to the calculation of the other mechanistic parameters:

$$k_{tot} = \frac{\sum_i k_i \, C_i \left(1 - e^{- k_i \, INTOH} \right)}{\sum_i C_i \left(1 - e^{- k_i \, INTOH} \right)}$$

Emissions weighing method "2" is analogous to method "1" except that it is appropriate for very slowly reacting species, where $k_i \times INTOH \ll 1$. For such species, the amount reacted is simply proportional to the OH radical rate constant, and the dependence on INTOH cancels out, making it unnecessary to estimate this quantity. If this method is chosen, then (following the notation used above), C_{tot}, and p_j are calculated using the equations,

$$C_{tot} = \frac{\sum_i C_i k_i}{k_{tot}}$$

and

$$p_j = \frac{\sum_i p_{ji} C_i k_i}{\sum_i C_i k_i}$$

which are the same as the equations shown above if k_i x INTOH <<1. As with method "1", k_{tot} is either input explicitly in the GROUP command, or, if not given there, is calculated using

$$k_{tot} = \frac{\sum_i C_i k^2_i}{\sum_i (C_i k_i)}$$

Note that the emissions weighing method affects not only the output of the PREPEMIT program in its calculation of the mechanistic parameters of the lumped groups, but also affects the lumped species assignments for the SAROAD classes used by the LUMPEMIT program when calculating molar emissions of the lumped group from the mass emissions of the corresponding SAROAD classes. For example, suppose a SAROAD class S, with mass emissions of E_S, is represented by a detailed model species A which has a molecular weight of mw_A and an OH radical rate constant of k_A. If model species A is represented by a lumped group L_0 using weighing method "0," i.e., with no rate constant weighing, then the contribution of emissions of S to the molar emissions of L_0 is simply given by

$$(\text{moles } L_0 \ / \text{ grams S}) = E_S/mw_A$$

and is independent of the OH radical rate constants of either A or L_0.

If, on the other hand, model species A is represented by lumped group L_1 which uses weighing method "1" and which has an OH rate constant k_1, then the contribution of S to molar emissions of L_1 is given by

$$(\text{moles } L_1 \text{ / grams S}) = \frac{(E_S/mw_A) \times (1 - e^{-k_A \text{ INTOH}})}{(1 - e^{-k_1 \text{ INTOH}})}$$

while if A is represented by lumped group L_2 which uses weighing method "2" and has an OH rate constant of k_2, then the contribution of emissions of S to that of L_2 is given by

$$(\text{moles } L_2 \text{ / grams S}) = \frac{(E_S/mw_A) \times k_A}{k_2}$$

where in both the latter two cases the contributions depends on the rate constants for both the detailed model species and the lumped group.

Note that if rate constant weighing of emissions is employed, then in general the number of moles of lumped groups emitted will not be the same as the number of moles of emitted detailed model species which are being represented. The PREPEMIT and LUMPEMIT programs uses the pseudo-model species "NROG" ("Non Reactive Organic Gas") to represent the difference between the number of moles of detailed model species actually emitted, and the number of moles of the lumped groups used to represent them. If the OH radical rate constant for the detailed model species is less than that for the lumped group representing it, then there will be fewer moles of the lumped group emitted than detailed model species, and the difference will be represented by emissions of the corresponding amount of the pseudo-species "NROG." If the OH radical rate constant for the detailed species is greater than that of its lumped group, then the corresponding number of moles of lumped species used will be greater than the moles of detailed species it represents, and emissions of a negative amount of "NROG" will be used to balance out the total number of moles emitted. (If the user wishes to avoid negative emissions of NROG, then rate constants for the lumped groups which use rate constant weighing can be assigned such that they are at the high end of the range of rate constants for the lumped group. However, if no rate constant is explicitly assigned for such lumped groups, and if the emissions profile is similar to the representative profile input into PREPEMIT, then the total amount of NROG emissions associated with such groups will be small.) PREPEMIT also calculates molecular weights for the lumped groups and NROG so that the total mass emitted for the representative profile is also conserved. The calculation of NROG emissions allows the airshed modeling to keep track of the total number of moles and mass emitted, regardless of which emissions weighing method is used.

3. Preparation Program Input for Lumped Groups

In order for the output of the PREPEMIT program concerning parameters for lumped groups to be properly processed by the airshed modeling program, the chemical mechanism incorporated in the airshed model must include reactions for the lumped groups which are given in the appropriate formats. In addition, the symbols used for the lumped groups, and the specifications of their reactions and mechanistic parameters, must be consistent with the symbols used for them in the lumping control and mechanistic parameter files. This means that there must be correspondences between the data in the lumping control and mechanistic parameter files input into PREPEMIT and the input used in the preparation program to define the reactions of the lumped groups when the mechanism is being prepared for use in the airshed model. These correspondences, and other considerations involved in preparing preparation input for lumped groups, are discussed in this section.

(It is assumed in this discussion that the airshed model is using a chemical mechanism which was prepared using the preparation program which is described in this document, though in principle it could be used with mechanisms prepared using other mechanism compilers or "hard wired" mechanisms if the necessary features are included.)

The considerations involved in model preparation input for lumped groups which represent alkenes are somewhat different than those used for alkanes and aromatics (or species lumped with them), and thus these will be discussed separately. For lumped groups representing species such as alkanes and aromatics, which are assumed in the mechanism to react only with OH radicals, and whose parameters are input into PREPEMIT either using "general" or "alkane" format parameter files, the preparation program input specifying their OH radical reaction must be in the following format:

```
.RXN
g1OH)            ;oh + nam1 = #g1y1 product1 + #g1y2 product2 + ...
g2OH)            ;oh + nam2 = #g2y1 product1 + #g2y2 product2 + ...
```

(See Section II.B.5 for a general discussion of the formats for reaction specification input into the preparation program.) The two reactions shown above give the reactions for two lumped groups, whose names are designated by the four-character codes "nam1" and "nam2," and whose parameter prefixes are the two character symbols "g1" and "g2," respectively. These codes and symbols must be the same as the symbols used for the name and the parameter prefix for the lumped group on the GROUP command in the lumping control file. "Product1," "product2," etc. refer to the various products, intermediates or sets of products and intermediates (either radical, stable, or pseudo-species chemical "operator") which could be formed in the reactions of the species represented by these lumped groups, and "y1," "y2," etc. are the two-character symbols used to identify yields of these products or intermediates in the mechanistic parameter files. The parameters "g1y1," "g1y2," "g2y1," etc. thus represent the yields for these various products or intermediates for this particular lumped group, which should be zero by default. These parameter names must not be decodable as numbers and the parameters should not be assigned non-zero values in the preparation program input. The symbol "oh" is the name used for OH radicals in the mechanism (which in the example mechanisms distributed with this software is always "HO."). The upper case and non-alphanumeric in the reactions above (other than the "..." notation) must be entered as shown.

The preparation program input defining the reactions of lumped groups which represent alkenes (whose parameters are input in "alkene" format parameter files) is somewhat different, because these can also react with ozone, NO_3 radicals, and $O(^3P)$ atoms, and also because a fixed nomenclature is used for the two-character mechanistic parameter symbols. The mechanistic parameter symbols generated by PREPEMIT for the alkenes (i.e., for detailed model species whose parameters are input using "alkene" format parameter files) are shown in Table 9. The preparation program input for a lumped alkene group with the name "ole1" and the parameter prefix "o1", should be in the following general form:

```
.RXN
 o1OH)          ;oh + ole1 = #o1ohp1 product1 + #o1ohp2 product2 + ...
 o1O3)          ;o3 + ole1 = #o1o3p3 product3 + ...
 o1N3)          ;no3 + ole1 = #o1n3p4 product4 + ...
 o1OA)          ;o3p + ole1 = #o1oap5 product4 + ...
```

where the symbols "oh," "o3," "no3," "o3p" are those used in the mechanism for OH radicals, ozone, NO_3 radicals, and $O(^3P)$ atoms, respectively, and the uppercase characters and non-alphanumeric symbols (other than "...") should be entered as shown.

Unlike the "general" or "alkane" lumped groups, the product yields for the reactions of the alkene lumped are not output directly by PREPEMIT, but instead PREPEMIT outputs various parameters (listed in Table 9) which can be used to derive the product yields. It is the responsibility of the airshed modeling program to derive these product yields given these parameter values. The Preparation program therefore must include the code such that the airshed modeling program will do this. This can be done using ".INS INIT" or ".#C1" input in the preparation program (see Section II.B.9). For example, the parameter "P1" indicates the number of =CH2 groups about the double bond of the alkene. Assuming that ethene is not represented by the lumped group, and thus considerations involving glycolaldehyde formation can be ignored, this means that (according to the current understanding of alkene reactions) the HCHO yield in the OH radical reaction should be given by P1 x (1 - PN), where (1-PN) is the fraction of OH radical reaction which does not react to form organic nitrates. If, for the lumped group "ole1" in the example above the symbol used for the HCHO yield in the OH radical reaction is "o1ohA1," then the code for calculating the appropriate value for this parameter from the PREPEMIT-supplied values of "o1P1" and "o1PN" can be generated by including the following inserted instruction input into the preparation program input:

Table 9. Mechanistic Parameters and Parameter Codes Output by PREPEMIT for Lumped Alkene Groups

Parameter Suffix Code	Mechanistic Parameter
	Codes for Substituents about double bond[a]
P1	Number of $=CH_2$ groups
P2	Number of $=CHCH_3$ groups
P3	Number of $=CHR$ groups (R not H or CH_3)
P4	Number of $=C(CH_3)_2$ groups
P4	Number of $=CR(CH_3)$ or CR_2 groups
	Other Mechanistic Parameters
PN	Organic nitrate yield in OH radical reaction
GA	Glycolaldehyde yield in OH radical reaction (used only if ethene is lumped with other alkenes)
NC	Number of carbons in alkene

[a] For dialkenes, the code should refer to the substituents about the bond where reaction is most likely to occur.

using "Pseudo-Machine" Format:

```
.#C1
L1#'1',S1#'o1PN',M1#'o1P1',E1#'o1ohA1'
```

or using "Pseudo-FORTRAN" Format:

```
.INS INIT
'o1ohA1' = 'o1P1' * (1.0 - 'o1PN')
```

Such input must be included for all the products for all four of the reactions of each of the alkene lumped groups in the mechanisms. A study of the preparation input files for the alkene lumped groups for the example mechanisms (see Section V on Appendix A) should make it clear the types of input which are required.

It is important that the reactions of all the lumped groups referenced by GROUP commands in the lumping control file be included in the mechanism used in the airshed model, and that all the lumped group names, lumped group parameter prefixes, and mechanistic parameter codes be referenced and defined properly in the mechanism. Otherwise, the airshed modeling program will either abort during the mechanism initialization process because of unrecognized input from PREPEMIT, or (worse) the program may run but give chemically invalid results. It is also advisable not to have lumped groups in the mechanism which are not referenced by GROUP commands in the lumping control file, or the mechanism will have species which will never be used in the airshed model simulation, which will make the calculation take more computer time and memory than necessary.

IV. DESCRIPTION OF SOFTWARE COMPONENTS

A. Model Preparation Program Software

1. Program Modules and Global Variables and Arrays

The model preparation program is written in FORTRAN-77, and has been implemented and tested on a DEC PDP-11/73 system running under the RSX-11M operating system, and on a DEC VAX computer running under VMS. Previous versions of this software have run on CDC Cyber systems running under the NOS operating system. It should be relatively straightforward to convert this software to run on other computers and operating systems that have a full-feature FORTRAN-77 or FORTRAN-V compiler.

The source code for the preparation program is organized into 18 files, which are listed in Table 10. The file PSPECS.FTN contains specification statements for the global parameters and common variables and arrays used throughout the program, and the other files consist of source code for the various program modules, including utility subroutines which are used by other SAPRC modeling programs besides just the preparation program. The names of the subroutines and program modules contained in the various source files are also listed in Table 10, along with a brief description of the function of the subroutine or module, and a list of the subroutines which they call.

The specification statements in the file PSPECS.FTN are used in most of the preparation program modules, other than the utility subroutines. The source code references the PSPECS file using the FORTRAN INCLUDE statement. Most of the specifications are for variables and arrays in COMMON blocks, and represents the primary means that data are passed from one module to another. Other variables and arrays used in this program are local to the various modules, and are not described in detail here. A description of the variables and arrays in the commons specified in PSPECS.FTN is given in Table 11.

The source files include a second file of specification statements, NEWIOBUF.FTN. This contains specifications for input/output parameters and buffers. It is referenced in PSPECS, and thus is used in all the preparation program subroutines, and also in the NEWINBUF input utility subroutines.

Table 10. Source Files and Program Modules for the SAPRC Model Preparation Program for Use with Airshed Models

.FTN File	Program Module(s)	Description	Subroutines Called[a]
		Specifications	
PSPECS	-	Specifications for global parameters and common variables and arrays for the preparation program. INCLUDE'd in all program modules except utility routines.	
NEWIOBUF	-	Specifications for I/O control parameters, variables and buffers. INCLUDE'd as part of the PSPECS specifications and in the NEWINBUF I/O utility routines.	
		Main Program	
PREP	MAIN	Initializes and opens input and listing output files. Reads in program control parameters. Calls routines to read, process, and save mechanism. Outputs program termination messages. input (.PRP) and listing output (.PRO) files.	ERRSET[b] RDRXN REORDR PNHRXN SAVERX
		Input Processing and Reaction Decoding	
RDRXN	RDRXN	Initializes mechanism variables and arrays. Inputs and processes model specification input. Builds up lists of species, coefficients, reactions, and kinetic parameters. Calls routines to decode reaction list. Does all processing of kinetic parameter input except for photolysis reactions. Writes reaction list input, inserted code input, and absorption coefficient, quantum yield (.PHOT) input onto temporary files for later processing by other modules. Handles conditional input control. Outputs listing of reactions to listing file as they are input.	RXLST1 RXLST2 SPCNAM
RXLST1	RXLST1	Called by RDRXN to process reaction lists. Adds new species and coefficients to list as they are encountered. Creates arrays giving indices of reactants, products,	COFNAM CONTLN SPCNAM

(continued)

Table 10 (continued) - 2

.FTN File	Program Module(s)	Description	Subroutines Called[a]
		and coefficients for each reaction. Does extensive error checking of reaction lists.	
RXLST2	RXLST2	Called by RDRXN to decode list of species when used in species specification input. Calls SPCNAM to process species names found on list.	SPCNAM
RXSUBS	SPCNAM	Called by RXLST1 or RXLST2 to determine if a species name has already been defined, and adds it to list if not. Sets species type based on value of IND, set by calling routine. If a new species is constant, gives it a default concentration of 1.	
	COFNAM	Called by RXLST1 process coefficient name, and assign it a coefficient index number. Tries to decode the coefficient name as a number, and if successful, sets this as a constant and checks to see if a constant coefficient of this value already exists. If it is not decodible as a number, checks to see if the input name matches a name on the list of variable coefficient names. Adds new coefficient to list of constant or variable coefficients if not found on list.	
	CONTLN	Called by RXLST1 to input continuation lines for reaction lists. Sets location pointer to first non-blank (or tab) character in line.	
		Processing and Listing of Species Input	
REORDR	REORDR	Called by MAIN after all input has been processed to order all species input according to their types, and verify that species are of the appropriate types based on whether they react are formed in a reaction. Species are ordered constant, buildup, active, and steady state. If a type is inappropriate (e.g., a species declared as "buildup" but which actually reacts), a warning	

(continued)

Table 10 (continued) - 3

.FTN File	Program Module(s)	Description	Subroutines Called[a]
		message is given and the type is changed. If a species is not found in any reaction, its type is changed to "dummy."	
LISTSC	LISTSC	Outputs list of species, sorted by type, to mechanism listing (.PRO) file	
		Mechanism Dependent Code Preparation	
PNHRXN	PNHRXN	Controls production of FORTRAN code for mechanism-dependent subroutines. Opens mechanisms .FTN file for output, and writes initialization code. If inserted instructions for initialization code are included, calls INSCOD or VCOEF to process them and output the code, depending on the format used. Inserted code instructions are read from the temporary file created by RDRXN. Calls BLDUP to produce the code for calculating rates of formation of "buildup" species. If inserted instructions for derivative calculation (DIFUN) code are included, calls INSCOD or VCOEF to process those instructions and output the code. Then calls DIFFUN to produce the mechanism-dependent code to calculate rates of change of active species due to chemical reaction. Closes mechanism's .FTN file when done.	BLDUP DIFFUN INSCOD VCOEF
BLDUP	BLDUP	Called by PNHRXN to produce FORTRAN code for calculating rates of formation of buildup species from the rates of reactions forming them.	
DIFFUN	DIFFUN	Called by PNHRXN to produce FORTRAN code used to calculate rates of change of active species. This includes producing code for calculating rates of reactions with no steady-state reactants; determining the order of steady state species for calculating their concentrations; producing code for calculating steady state species	

(continued)

Table 10 (continued) - 4

.FTN File	Program Module(s)	Description	Subroutines Called[a]
		concentrations; producing code for calculating rates of reactions of steady state species; and finally producing code for calculating the rates of formation and consumption of the active species.	
INSCOD	INSCOD	Called by PNHRXN to decode inserted instructions written in the pseudo-FORTRAN format, and produces FORTRAN code corresponding to them.	
VCOEF	VCOEF	Called by PNHRXN to decode inserted instructions written in the pseudo-machine language format, and calls various entry points in ASMD to produce FORTRAN code corresponding to them	ASMD
ASMD	ASMD	Called by VCOEF to produce FORTRAN code to correspond to machine instructions. Has separate entry points which correspond to various machine instructions, such as Load, Store, Multiply, etc.	
SAVERX	SAVERX	Opens the model data (.MOD) file and writes all mechanism-dependent data (species and coefficient names, coefficient values, kinetic parameters, reaction lists, etc.) Reads files containing photolysis absorption coefficient and quantum yield data (either temporary files created by RDRXN to contain data following ".PHOT" commands, or files by the "PF=filename" kinetic parameter option), processes them for solar photolysis rate calculations, and saves the processed data on the .MOD file. Closes the .MOD file when done, and outputs summary of mechanism data in output listing (.PRO) file.	PFALIN
		Utility Subroutines	
PFALIN	PFALIN	Utility subroutine called by RDRXN to align absorption and quantum yield arrays according to the standard set	

(continued)

Table 10 (continued) - 5

.FTN File	Program Module(s)	Description	Subroutines Called[a]
		of wavelengths used for atmospheric photolysis rate calculation. Figures averages of absorption coefficient x quantum yield product for each of the standard wavelength intervals.	
NEWINBUF	INBUF	Reads an input record from the current input file. If the input record is "@filename," opens that file for input, keeping the previous input file open, then reads an input line from there. If the input record starts with a "!," another record is read. If an end of file is reached, closes input file and goes back to previous input file, if any. If no previous input file, sets an end-of-file flag, and returns to calling routine. Otherwise, passes input record to calling routine via IOB160 character array in common. Also decodes and acts upon "@#SHOW" and @#NSHOW" commands if encountered in the input record.	
	INITIN	Called to initialize INBUF processing. Argument tells how many input files can be open at once.	
	FILEIN	Closes all input files (if any are open) and causes input to come from file whose name is in IOB160.	
	FILNXT	Called to simulate the effect of a "@filename" input record, where the file name to be open for input is in IOB160. (Not used by the preparation program).	
NEWSUBS	TMARK	Returns time of day and date in a CHARACTER*16 argument. (This subroutine will probably have to be changed when implementing on other computers.)	

(continued)

Table 10 (continued) - 6

.FTN File	Program Module(s)	Description	Subroutines Called[a]
	various	Various string manipulation subroutines. Subroutines include ALIN8, ALIN16, FILNAM, MOVRIT, MOVLFT, and MOVLF8, and functions include LENFNM, NEQ16, and LENSTR. See source listing for the use of these utility subroutines.	

[a]Excludes calls to utility subroutines in NEWINBUF or NEWSUBS.
[b]ERRSET is a FORTRAN error control routine. It is called to suppress "input conversion error" error messages, which would otherwise result when variable coefficient names are being processed to determine if they are decodible into a number. This call may need to be changed for use on non-DEC computers.

(Some FORTRAN compilers do not have an INCLUDE feature. In this case, it is recommended that the programmers responsible for converting this software to run on their system use a source code pre-processing program to perform this function. The source code for a program which does this on CDC Cyber systems is available from the authors. It is not recommended that the programmers manually insert these specifications into the source code for each of the subroutines which use them, since this makes program maintenance significantly more difficult.)

2. Format of Mechanistic Data (.MOD) Files

The two major output files produced by the preparation program for implementing the chemical mechanism into an airshed model are the FORTRAN source files for the mechanism specific subroutines and the mechanistic data (.MOD) files. The .MOD files contain all the data required for the airshed modeling to implement the chemical mechanism and utilize the mechanism specific subroutines. In particular, the mechanism specific subroutines do not contain any DATA statements or BLOCK DATA routines, so it is necessary that the data in the .MOD files be input during the initialization process of the airshed modeling program. In this section, the organization of the mechanism specific data in these files are discussed.

Table 11. Global Parameters, and Variables and Arrays in COMMON Blocks Used by the Preparation Program

Name	Type and Size[a]	Description[b]
	Specifications for Species	
MAXNS	Integer Parameter	Maximum number of species (140)
MAXSS	Integer Parameter	Maximum number of steady state species (100)
NS	I*2	Number of species
NSC	I*2	Number of constant species
NB	I*2	Number of buildup species
NDUM	I*2	Number of declared dummy species
NDY	I*2	Number of active species
NSS	I*2	Number of steady state species
NDEL	I*2	Number of deleted dummy species
NSC1	I*2	First constant species no. Always 1
NSC2	I*2	Last constant species no. Always NSC
NB0	I*2	First buildup species no.-1. Always NSC
NB1	I*2	First buildup species no. Always NSC+1
NB2	I*2	Last buildup species no. Always NSC+NB
ND1	I*2	First dummy species. Always ND1=NB2+1
ND2	I*2	Last dummy species. Always NB2+NDUM
NO	I*2	First active species no.-1 Always ND2
NA1	I*2	First active species no. Always ND2+1
NO1	I*2	Same as NA1
NA2	I*2	Last active species no. Always ND2+NDY
NS1	I*2	First steady state species no. Always NA2+1
NS2	I*2	Last steady state species no. Always NA2+NSS
NAME	C*16 (MAXNS)	Species name
SPTYP	I*2 (MAXNS)	Species type index: 1 = constant; 2 = buildup; 4 = active; 5 = steady state; 6 = unspecified
CONCO	R*4 (MAXNS)	Default initial concentration
MWT	R*4 (MAXNS)	Default molecular weight
CNO	R*4 (MAXNS)	Default number of carbons
NNO	I*2 (MAXNS)	Default number of nitrogens
	Specification for Reactions and Kinetic Parameters	
MAXRXN	Integer Parameter	Maximum number of reactions (400)
MAXPRD	Integer Parameter	Maximum number of products and product coefficients (6*MAXRXN)
MAXKBF	Integer Parameter	Maximum number of kinetic parameters (1080)

(continued)

Table 11 (continued) - 2

Name	Type and Size[a]	Description[b]
MAXRCT	Integer Parameter	Maximum number of reactants per reaction (always 3)
NRXN	I*2	Number of Reactions
LOCKBF	I*2	Number of kinetic parameters in the KPBUF array
RXNLBL	C*4 (MAXRXN)	Reaction Label
RXTYP	I*2 (MAXRXN)	Type of reaction kinetics: 0 = SAMEK type; 3 = CONST type; 4 = extended Arrhenius thermal; 5 = FALLOFF thermal; 7 = photolysis reaction
LKBUF	I*2 (MAXRXN)	Use depends on type of reaction kinetics. For types 3, 4, and 5, gives index for first kinetic parameter in KPBUF array. For type 0, gives reaction number whose rate constant is used. For type 7, gives number of photolysis set used.
KPBUF	R*4 (MAXKBF)	Kinetic parameters for non-photolysis reactions. Number and use of parameters depends on type of kinetics, as follows: type (3): rate constant; type (4) A, Ea, and B; type (5): A0, E0, B0, AI, EI, BI, F, and N. First parameter for given reaction pointed to by LKBUF array. Not used for type 0 and 7 reactions.
NRTOSR	I*2 (MAXRXN)	Number of reactants in the reaction
IRTOSR	I*2 (MAXRCT,MAXRXN)	Species or coefficient numbers for reactants (up to 3). Species numbers are positive, coefficient numbers negative
NPRODS	I*2 (MAXRXN+1)	Index number for first product or coefficient index in IPRODS array. NPRODS(NRXN+1)-1 is last index no.
IPRODS	I*2 (MAXPRD)	Species or coefficient numbers for products. Species numbers are positive, coefficient numbers negative. NPRODS array used to locate product list for a given reaction.

(continued)

Table 11 (continued) - 3

Name	Type and Size[a]	Description[b]
	Specifications for Coefficients	
MAXCO	Integer Parameter	Maximum number of coefficients (480)
MAXCOV	Integer Parameter	Maximum number of variable coefficients. Always less than MAXCO, since the last MAXCO-MAXCOV coefficients are the constant coefficients. Note that MAXCOV+1 is always the index number for the first constant coefficient regardless of how many variable coefficients are in the mechanism. (420)
NCOCO1	Integer Parameter	Index number for the first constant coefficient. (Always MAXCOV+1)
NCOEFV	I*2	Number of variable coefficients
NCOC2	I*2	Index number for the last constant coefficient.
COEFNM	C*8 (MAXCOV)	Variable coefficient names
COEF	R*4 (MAXCO)	Coefficient values. First NCOEFV are values of the variable coefficients.
	Specification of Photolysis Data	
MAXPHK	Integer Parameter	Maximum number of photolysis sets (20)
MAXAQY	Integer Parameter	Maximum number of wavelengths which can occur in input photolysis sets (150)
NSOLWL	Integer Parameter	Number of wavelengths in photolysis sets after being processed for calculation of solar photolysis rates. This is the number of absorption coefficient x quantum yield values which are saved in the .MOD file. Internal to SAVERX subroutine. (always 27)
NPHOTK	I*2	Number of photolysis sets
PHOTNM	C*8 (MAXPHK)	Photolysis set name
PFTYPE	L*1 (MAXPHK)	Set to .TRUE. if photolysis set read from a the input stream with the .PHOT command, or set to .FALSE. if set to be read from a file referenced by the PF=filename kinetic parameter option.
IPHOTR	I*2 (MAXPHK)	Index giving number for first reaction which uses this photolysis set to derive its rate constant.

(continued)

Table 11 (continued) - 4

Name	Type and Size[a]	Description[b]
	Program Control Parameters and Variables	
MODFIL	C*16	Mechanism name. Used to name .PRP input file, and .FTN and .PRO output files.
TITLE	C*80	Title for mechanism. First 16 characters is date and time mechanism prepared, rest is title read as first non-comment record in .PRP file.
TEMPR	R*4	Default temperature for thermal rate constants. Default is 298.12 K, but can be changed during parameter input. Included in .MOD file
TREF	R*4	Reference temperature for thermal kinetic parameters. Default is 300 K, but can be changed during parameter input. Included in .MOD file output.
LITTLE	R*4	Minimum pseudo first order rate of destruction of all steady state species. must be a small, non-zero value to avoid possible divide-by-zero errors at the beginning of the simulations. A coefficient called "LITTLE" is created and given this value. Default is 1.0E-30, but this can be changed during parameter input.
INDLIL	I*2	Index for the special coefficient called "LITTLE." Always 1 in this version.
EOF	L*1	.TRUE. if end of file on input
ASMPRT	L*1	Used for debugging. Usually .FALSE. Almost all of its features are disabled in this version.
VCOFDF	L*1	.TRUE. if temporary file has been opened to save inserted instruction data. .FALSE. if no inserted instructions input
VCOCAL	L*1 (8)	Determines whether various types of inserted instructions need to be processed. Not all 8 types used in this version. Types used (and corresponding inserted instruction input command): 1 = ".#C1"; 2 = ".#D1"; 4 = ".INS INIT"; 5 = ".INS DIFF"

(continued)

Table 11 (continued) - 5

Name	Type and Size[a]	Description[b]
YESPRP	L*1	.TRUE. unless preparation aborted due to error
NOPRP	L*1	Opposite of YESPRP
NERR	I*2	Number of errors found
IND	I*2	Used for communicating between some subroutines. Communicates species type between RDRXN, RXLST1, RXLST2, and SPCNAM.
MODUIC	C*40	Prefix .MOD filename. Set to blank in this version, but can be given system-specific device or file directory name if .MOD files are to go into directories or devices other than the default
PHFUIC	C*40	Prefix for filenames to use when opening photolysis data (.PHF) files referenced by "PF=filename" kinetic parameter option. See discussion of MODUIC.
LOGUIC	C*40	Prefix for filenames for .PRP and .PRO files. See discussion of MODUIC.
TMPUIC	C*40	Prefix for used for temporary files. See discussion of MODUIC. Can be set to the name of a RAM drive on systems which have this hardware for faster program execution.
	Input/Output Control Variables and Parameters[c]	
IKEY	Integer Parameter	Unit number for terminal input (5)
ICRT	Integer Parameter	Unit number for terminal output (5)
OUT	Integer Parameter	Unit number for listing output (.PRO) file (9)
UNIT1	Integer Parameter	Unit number for file I/O (7)
UNIT2	Integer Parameter	Unit number for file I/O (8)
UNIT3	Integer Parameter	Unit number for file I/O (10)
IN	I*2	Current input unit (set by INBUF)
IOLEN	I*2	Length of input string (set by INBUF)
IORET	I*2	INBUF return code. 0 = normal return, 2 = end of file
LOUTBF	I*2	Length of output buffer

(continued)

Table 11 (continued) - 6

Name	Type and Size[a]	Description[b]
IOB160	C*160	Work buffer. Used to pass input from INBUF, communicating with some utility routines, and as a character scratch buffer.
BYT160	C*1 (160)	EQUIVALENCE'd to IOB160
IOBF16	C*16	Work buffer
BYT16	C*1 (16)	EQUIVALENCE'd to IOBF16
OUBUF	C*132	Work buffer used for listings output

[a]Species types: I*2 = INTEGER*2, R*4 = REAL*4, C*n = CHARACTER*n L*1 = LOGICAL*1. Size refers to maximum dimensions of arrays, which are given in parentheses if the symbol is an array.
[b]Value in parentheses is value of parameter in current version of the program. These can be changed by editing PSPECS.FTN.
[c]The specifications for all the I/O control variables and parameters except for OUT, and UNITn are in the file NEWIOBUF.FTN.

Only one .MOD file is produced for each prepared chemical mechanism processed by the preparation program. These are sequential access ASCII files with variable record length, but with no record longer than 80 characters. All the data in this file are written by the SAVERX subroutine, and the source code for that subroutine (in the file SAVERX.FTN) can be examined to see the exact WRITE statements made to this file. (The file is open and written to FORTRAN I/O unit labeled UNIT1.) The organization of the data are also summarized in Table 12. The variable names used in Table 12 refer to the internal variable names which are listed in Table 11.

3. Mechanism Dependent Subroutines Produced

The preparation program outputs the source code for mechanism dependent subroutines in the mechanism's .FTN file. This file contains the source code for three subroutines, called CONSTR, BLDUP and DIFUN. All of the communication of data between these subroutines and the calling program is through arguments, no COMMON blocks are used, and the subroutines perform no input or output operations. All arguments to the three subroutines are REAL arrays, containing data for the species, reactions, and coefficients in the ordering consistent with the output of the

Table 12. Organization of Mechanisms Specific Data on the .MOD Files Produced by the Preparation Program

Group	FORTRAN Format Code	No. Records[a]	Variables or Description[b]
1	(A8,' = MODEL FORMAT')	1	File format code. Should be 'MV:1/88A' for this format.
2	(A16, A65)	1	Date and time mechanism was prepared; mechanism label. (Both in TITLE)
3	(12I6)	1	Number of species (NS); number of active species (NDY); number of constant species (NSC); number of buildup species (NB); number of dummy species (NDUM); number of reactions (NRXN); number of variable coef's (NCOEFV); index no. for first constant coefficient (NCOC1); index no. for last constant coefficient (NCOC2); number of photolysis sets (NPHOTK); number of kinetic parameters (LOCKBF); and number of wavelengths for data in photolysis sets (NPHOTK)
4	(2F10.2)		Default temperature (TEMPR) and reference temperature (TREF)
5	(A16, F7.2, F6.2, I3, 1PE11.3)	(NSC+NB+ NDUM+NDY)	Species name (NAME); default molecular weight (MWT); default number of carbons (CNO); default number of nitrogens (NNO); and default initial concentrations for the constant, buildup, dummy, and active species, respectively.
6	(5A16)	(NS-NSC-NB -NDUM-NDY) /5	Species name (NAME) for the steady state species. No input if no steady state species in mechanism.
7	(4(A8, 1PE11.3,1X))	NCOEFV/4	Name; value for all variable coef's. (COEFNM(I), COEF(I), I=1,NCOEFV)
8	(1P6E12.3)	(NCOC2- NCOC1+1)/6	Values for all constant coef's. (COEF(I), I=NCOC1, NCOC2)

(continued)

Table 12 (continued) - 2

Group	FORTRAN Format Code	No. Records[a]	Variables or Description[b]
9	One Group of 4 records for Each of the NPHOTK Photolysis Sets		
9a	(A8, I5, 7X, 1P6E10.3	1	Photolysis set name; index for first reaction using this set; absorption coefficient x quantum yield values for the first 6 wavelengths. PHOTNM(I), IPHOTR(I), (AQ(I),I=1,6),
9b	(8E10.3)	3	Absorption coefficient x quantum yield values for rest of the NSOLWL wavelengths, where NSOLWL = 27. (AQ(I), I=7,27)
10	(1P8E10.3)	LOCKBF/8	Kinetic parameter data. See Table 11. (KPBUF(I), I=1,LOCKBF)
11	(16I5)	NRXN/16	Kinetic parameter index for each reaction. See Table 11. (LKBUF(I), I=1,NRXN)
12	(16I5)	NRXN/16	Kinetic parameter type for each reaction. See Table 11. (RXTYP(I), I=1,NRXN)
13	(10A8)	NRXN/8	Reaction labels. (RXNLBL(I), I=1,NRXN)
14	(L1, 1X, A78)	NRXNS+c (c)	Logical variable indicating whether there are additional input for this reaction (.TRUE. if none, .FALSE. if additional input for this reaction); part or all of reaction list.

[a]If an expression is given which evaluates to a non-integer, it is rounded up to the next higher value. If expression evaluates to zero, there is no input for this group.
[b]See Table 11 for a description of the variables written onto this file. Variables listed here which are not listed in Table 11 are local variables in WRITRX.
[c]"c" is total number of continuation lines for all reactions in the mechanism.

preparation program. (The mechanism dependent subroutines do not need to be passed information concerning the numbers of species, reactions, etc., since that information is already "hard wired" into them.) A consistent nomenclature for these arrays are used for all three subroutines. The purpose and operations of these subroutines, and the arrays they expect as arguments, are discussed in the sections below.

a. CONSTR Initialization Subroutine

The purpose of this subroutine is to initialize values of parameters, initial species concentrations and rate constants. It should be called by the airshed modeling program after all of the parameters and initial reactant concentrations have been read in, but before the model simulation begins. The preparation program does not automatically produce any code for this subroutine; it consists entirely of inserted instructions explicitly included in the model preparation input files. Such instructions might be utilized if, for example, some coefficients, rate constants, constant species concentration or initial active species concentration are to be determined based on values of other coefficients, rate constants, constant species concentration or initial active species concentration. Inserted instructions are discussed in Section II.B.9, above, where several examples of their application are given. The inserted instructions to go into the CONSTR subroutine are those following the ".#C1" or the ".INS INIT" commands. If no such commands are included in the preparation program input data, then the CONSTR subroutine has no operations, other than just to return control to the calling routine.

(The reason this subroutine is called CONSTR is because initially it was used to merge the constant reactant concentrations with the rate constants, so those multiplications did not have to be carried out at each time step. However, this feature was disabled because frequently applications were found where it was useful to have the rate constants of reactions involving constant species change during the calculation, or to carry out multiple sets of calculations where constant species concentrations may vary. This was incompatible with the practice of automatically merging constant concentrations and rate constants. If this feature is desirable, it can be explicitly added by using appropriate inserted initialization instructions in the model preparation input.)

The four arrays referenced as arguments to CONSTR, given in the order they appear on the argument list, are:

1) The array "C," containing the constant species concentrations;
2) The array "A," containing the active species concentrations;
3) The array "RK," containing the reaction rate constants; and
4) The array "CO," containing the coefficient values.

These arrays all contain values initially, but they can be modified by CONSTR as a result of the inserted instructions specified for inclusion in CONSTR. Note, however, that for airshed model calculations, where, in general, there may be different initial active species concentrations in each cell of the model, the use of inserted instructions to modify or reference the "A" array is not recommended, unless an appropriate code to take this into account is included in the calling routine. However, the ability to reference active species concentrations is retained for compatibility with mechanisms used for environmental chamber simulations, since it is used in some chamber effects representations.

b. DIFUN Derivative and Rate Calculation Subroutine

The DIFUN subroutine is the main mechanism dependent subroutine produced by the preparation program. This is the subroutine the airshed modeling program calls at each time step for the purpose of calculating the rates of change of the integrated (active) species due to chemical reaction. This subroutine has eight real array arguments, listed below in the order they appear on the argument list:

1) The array "C," containing the constant species concentrations (input);

2) The array "A," containing the active species concentrations (input);

3) The array "S," containing the concentrations of the steady state species (output);

4) The array "RK," containing the reaction rate constants (input);

5) The array "R," containing the values of the rates of each of the reactions (output);

6) The array "CO," containing the coefficient values (input);

7) The array "FR," containing the rates of formation of the active species due to chemical reaction (output); and

8) The array "LR," containing, for each active species, a pseudo-first order rate constant giving its overall rate of consumption due to chemical reaction (output).

As indicated above, the contents arrays C, A, RK, and CO are expected to be defined when the program is called and are not normally changed by DIFUN (unless there are inserted instructions added to the mechanism for coefficients which depend on other coefficients or concentrations that need to be updated at each time step), while the contents of the arrays S, R, FR, and LR are defined by DIFUN. Any values in these output arrays at the time DIFUN is called are ignored and over-written.

The output arrays S and R can be treated as scratch arrays by the calling program, unless (1) the concentrations of the steady state species are to be saved or output, or (2) it is desired to calculate the rates of formation of the non-reacting (buildup) species. In the former case, the array S must be saved, while in the latter case the rates of the reactions of the "R" array are required as input to the BLDUP mechanism specific subroutine, discussed below. However, even if the steady state species concentrations or the rates of the reactions are not required, these arrays must be included as arguments to DIFUN, and sufficient space must be allocated to them to hold the concentrations of all the steady state species (for S) or the rates of all the reactions (for R). If there are no steady state species, the array S is not used, but it is still included on the argument list.

The output arrays FR and LR are used to calculate the rates of change of the active species due to chemical reaction. The derivatives of the active species due to chemical reaction are not calculated by DIFUN, but they can be readily calculated, for each active species i, by

dA(i)/dt (due to chemistry) = FR(i) - A(i) x LR(i).

where A(i) is its concentration at time t. The rates due to chemical reaction are output in terms of FR and LR to be compatible with the "hybrid" integration technique employed in the airshed model software currently in use by the ARB.

Note that the separate calculation of FR and LR involves more operations in DIFUN than just calculating the total rates of change of the active species due to chemical reaction. Thus, if a solver (such as LSODE) is used which only requires the total rate of change, and does not require separate calculation of FR and LR, the DIFUN code should be modified accordingly for more efficient operation. Other versions of the preparation program at SAPRC calculate only the total rate of change, so it would not be difficult to modify this version of the preparation program if a solver which does not require separate calculaton of FR and LR is to be employed. Since DIFUN is called many times during the execution of the airshed model, it is important that it be as efficient as possible.

The preparation program input options ".#D1" or ".INS DIFF" can be used to have inserted instructions added to the DIFUN code. These instructions are added to the first portion of the code, and thus if they change rate constants, coefficient values, or constant species concentrations, these changes are made before they are used in any other calculations, such as, for example, calculating rates of reactions or rates of formation of active species. Note that the inserted instructions should NOT change the values of the active species concentrations, though the preparation program does not check for or prevent this.

The operations of the DIFUN subroutine are carried out in several steps, summarized briefly as follows: (1) The inserted instructions, if any, are executed. (2) The rates, or partial rates, of all reactions are calculated by multiplying all the rate constants by the concencentration of the non-steady state reactants and rate coefficients and placing them in the R array. (3) The concentration of each steady state species, starting with one whose concentration depends on that of no other steady state species, is calculated. The rates of all the reactions forming it are summed up, and this is divided by the sum of the coefficient "LITTLE" (see Section II.B.2) plus the partial rates of the reactions consuming it, and the resulting value is placed in the S array. (4) Immediately after the concentration of each steady state species is calculated, the rates of all its reactions are calculated by multiplying their partial rates (in the R array) by the steady state species concentration, and placing the full rate back in the R array. (5) The formation rates of each active species are calculated by summing up all rates of all reactions forming it, multiplied by product coefficients, if appropriate, and placing the results in the FR array. (6) Finally, the loss rates of each active species are calculated by summing up the partial rates of each reaction the species undergoes (multiplying the rate constant by the concentrations of the other reactants, and by the number of times the active species appears as a reactant in the reaction, if more than once), and placing the results in the LR array.

c. BLDUP Subroutine for Formation Rates of Non-Reacting Species

Most chemical mechanisms include species which are formed in a reaction but do not undergo subsequent reaction. These are refered to in this document and in the comments in the source listing as "buildup" species. Since their concentrations do not affect any other species concentration, calculations of buildup species concentrations is optional. Thus the mechanism specific code needed to calculate their concentrations is included in a separate subroutine, called BLDUP. This subroutine need not be included in the airshed modeling program if calculation of concentrations of buildup species is not desired. However, if they are desired, this subroutine should be called at each time step, or at least as frequently as necessary to obtain reasonably accurate concentrations for the non-reacting species.

The BLDUP subroutine takes as input real arrays giving the instantaneous rates of each of the reactions and the stoichiometric (and other) coefficients, and produces as output a real array giving the rate of formation, due to chemical reaction, of each of the buildup species. Since the rates of reaction are required as input to BLDUP, the calling program must, at each time step, call DIFUN first, so that these rates will be defined when BLDUP is called. The specific arguments used, and the order they appear on the argument list, are as follows:

1) The array "R," to contain the values of the rates of each of the reactions (input);

2) The array "CO," containing the coefficient values (input); and

3) An array "DY," to contain the calculated rates of formation of the bldup species (output).

None of the contents of the R and CO are changed by this subroutine. (Inserted subroutines cannot be added to BLDUP.) Any values previously in "DY" are ignored and over-written by BLDUP. The DY array must contain one location for each buildup species in the mechanism. No scratch arrays or other storage allocation is needed for the operation of this subroutine.

Note that if accurate calculation of a subset of the non-reacting species are of particular importance for an airshed model application, the user has the option of having selected non-reacting species being explicitly integrated, i.e., being treated as active. This is done by declaring the species of interest as active by using the ".ACT" option in the preparation program input file. (This and other species specification options is discussed in Section II.C.) The program will then include the calculation of the rate of formation of the species in the DIFUN subroutine (and not in BLDUP) just as it does with any other active species, despite the fact that it doesn't react. In that case, the DIFUN will always set the element of the LR array corresponding to that species to zero.

d. Memory Allocation Considerations

When used in multi-celled airshed models, it is recommended that the arrays of rate constants (RK) and coefficient values be the same for all the cells used in the model; i.e., that they not be allocated as separate arrays for each cell. Otherwise, the net effect would be that different chemical mechanisms would be used in different cells, which means that chemical properties of species would be changing as they move from one cell to the next. In addition, the arrays for steady state species concentrations (S), reaction rates (R), buildup species derivatives (DY), and active species formation and loss rates (FR and LR) can be used as temporary work arrays and need not be saved from one step to the next, or from one cell to the next. Thus the amount of memory allocated to these arrays (i.e., the number of reactions, coefficients, and steady state species) should not be a major factor in affecting the total memory requirements of the airshed model. On the other hand, obviously the active species concentrations need to be saved from step to step, and separate allocation is required for each of the cells used in the airshed model. In addition, if the concentrations of the buildup-only species are to be calculated (i.e., if the BLDUP subroutine is to be used), memory needs to be allocated for the concentrations of each of these species for each of the cells.

Note that the locations in the rate constant array which correspond to reactions whose kinetic parameters are specified with the SAMEK option (see Section II.B.5.a) are not used, since the mechanism dependent code in the DIFUN subroutine (discussed below) use the rate constant for other reactions. Nevertheless, the code requires that those locations be allocated.

Note also that the coefficient value array is organized such that the first NCOEFV values correspond to the variable coefficients and the values in the locations NCOC1 and NCOC2 contain the values of the constant coefficients. (The values of NCOEFV, NCOC1, and NCOC2 are read from the .MOD file, along with the initial values of the coefficients in the coefficient array.) The mechanism dependent code in all three of these subroutines require that the coefficient values in the "CO" conform exactly to this organization, i.e., that the variable and the constant coefficients be in the same array. Thus, in general, there will be unused locations in the coefficient value array between NCOEFV and NCOC1, but the code requires that these locations be allocated. The number of wasted locations in the coefficient value array can be minimized by reducing the MAXCOV parameter in the preparation program to a value just slightly higher than the actual number of variable coefficients. However, since it is only necessary that one coefficient and rate array be allocated for the entire model, in practice the total amount of wasted memory should be insignificant. If the capability of having separate variable coefficient values for each cell are desired, the preparation program code should be modified so that separate arrays be used for variable and constant coefficient values.

B. Emissions Processing Programs

As discussed in Section III, this software system includes two programs, designated PREPEMIT and LUMPEMIT, for pre-processing of emissions input into the airshed model. Both of these programs are written in FORTRAN-77, and have been developed and tested using a PDP-11/73 microcomputer using the RSX-11M+ operating system. It should be relatively straigntforward to convert these programs to run on other systems which have full-featured FORTRAN-77 or FORTRAN-V compilers. The software components, internal data structures, and other programming aspects of these two programs are discussed in this section. The PREPEMIT and the LUMPEMIT programs are discussed in Sections IV.B.1 and IV.B.2, respectively, and the organization of the data sets produced by these programs are discussed in Section IV.B.3. The use of these programs and the formats of their input data sets have been discussed in Section III.

1. PREPEMIT Program Modules and Global Variables and Arrays

The PREPEMIT program is intended to be a complete program to prepare emissions assignments and other emissions-related initialization data needed for airshed model runs using a given chemical mechanism, emissions "lumping" scheme, and representative emissions profile. If the appropriate interface modules are included in the emissions processing program (e.g., LUMPEMIT) as discussed in the following section, and in the airshed model programs as discussed in Section IV.C.3, then PREPEMIT should be able to be used without extensive modifications. Nevertheless, it is useful that the programmer know the functions of the various program modules used by PREPEMIT, and the internal data structures that they employ. These are briefly summarized in this section. In addition, the source code itself contains comments which should help the programmer understand the functions of the modules and factors they should consider if changes to this software are desired.

The source code for the PREPEMIT program is organized into 11 files, which are listed in Table 13. These consist of two files containing specifications statements for the global parameters and common variables, and nine files containing the source code for the program modules. The names of the PREPEMIT program modules in these nine files, and a brief description of each, are also included in Table 13.

The two specification files used by PREPEMIT are LUMPSPEC.FTN and SARSPEC.FTN. The former contains specifications for global parameters and common variables which are used by all the modules of PREPEMIT. All program modules reference the LUMPSPEC.FTN file using the fortran INCLUDE statement. The common blocks specified in LUMPSPEC are the major means of passing data from one module of PREPEMIT to another. This includes most of the data read from the lumping control and mechanistic parameter files,

Table 13. Source Files and Program Modules for the PREPEMIT Program

.FTN File	Program Module(s)	Description	Subroutines Called
		Specifications	
LMPSPEC	-	Specifications for global parameters and variables used by all modules.	
SARSPEC	-	Specifications for data read from the SAROAD.PRM data set. Used by LDSAR, LMPEMI, and SARSUB	
		Program modules	
PREPEMIT	MAIN	Controls overall operation of the program. Calls RDLUMP to read lumping control data, LDSAR to read SAROAD.PRM data, and performs other initializations. Gets name of data set containing representative emissions profile, and calls LMPEMI to read it. Gets name of data sets for lumped model species parameters and levels, and calls LMPSUB to produce and output these. Gets name of SAROAD assignment data set and calls SARSUB to produce and output it.	RDLUMP LDSAR LMPEMI LMPSUB SARSUB
LDSAR	LDSAR	Opens SAROAD.PRM and loads its data into the arrays specified in SARSPEC.FTN. All the data necessary to derive detailed model species compositions from mass emissions of SAROAD classes are read in or initialized.	
RDLUMP	RDLUMP	Initializes parameters for lumped groups, lists of explicit model species. Asks user for name of lumping control file, and reads and processes lumping control commands. Calls RDGENP, RDALKP, or RDOLEP to read data in mechanistic parameter files referenced by PARM commands. Opens and reads files referenced in the REPLACE or SPECIES commands, if any. All the data necessary to derive lumped model species parameters and levels from detailed model species compositions are read in or initialized by this module or those it calls.	RDGENP RDALKP RDOLEP

(continued)

Table 13 (continued) - 2

.FTN File	Program Module(s)	Description	Subroutines Called
RDGENP	RDGENP	Called by RDLUMP to read mechanistic parameter files in the "general" format	
RDALKP	RDALKP	Called by RDLUMP to read mechanistic parameter files in the "alkane" format	
RDOLEP	RDOLEP	Called by RDLUMP to read mechanistic parameter files in the "alkene" format	
LMPEMI	LMPEMI	Reads an input emissions file giving the representative emissions profile in terms mass emissions of SAROAD classes, and uses the data read by LDSAR from SAROAD.PRM to convert this profile into molar emissions of detailed model species.	
LMPSUB	LMPSUB	Given the molar emissions of detaled model species and the lumping control data input by RDLUMP (and the modules it calls), this module calculates the molar emissions of lumped model species and the parameters for the lumped groups which correspond to this emissions profile. It then outputs the parameters and other data for the lumped model species to the lumped species parameter file, and outputs the lumped species composition of the emissions profile either to the parameter file or to a separate file, depending on the arguments passed by the calling program. These output file(s) are opened by the calling routine.	
SARSUB	SARSUB	Produces the output file giving the lumped species assignments for the SAROAD classes, which is intended to be read by the LUMPEMIT program. Uses the data in the SAROAD.PRM file read by LDSAR, the lumping control data read by RDLUMP (and the modules it calls), and the rate constants assigned to the lumped groups calculated by LMPSUB to produce this file. The output file is opened by the calling routine.	

and most of the processed data for the representative emissions profile and the lumped model species. The major exceptions are the specifications for the detailed model species composition of the representative emissions profile, and the specifications for the data in SAROAD.PRM. The specifications for the detailed model species compositions are given in the MAIN routine and are passed to the two subroutines which need them as arguments. The specifications for the data in SAROAD.PRM are given in the file SARSPEC.FTN, and are referenced using the FORTRAN INCLUDE statement by the three modules (LDSAR, LMPEMI, and SARSUB) which use them. The global parameters, variables, and arrays which are shared by the various modules of PREPEMIT are listed in Table 14, along with a brief indication of their use. The table also indicates where the specifications for these data are located.

2. LUMPEMIT Program Modules and Data Organization

The LUMPEMIT program is intended to be a simplified representative of a program which prepares emissions data for input to an airshed model run, based on the output of PREPEMIT. Almost certainly some modifications to LUMPEMIT would have to be made for it to be compatible with existing emissions data bases and airshed model software systems. Alternatively, the appropriate program modules in LUMPEMIT could be incorporated into existing emissions pre-processing software systems. In particular, the operation of LUMPEMIT is not designed for efficient processing of extremely large numbers of emissions profiles, which may in general be necessary for airshed model runs. However, the internal operation of LUMPEMIT is relatively simple, and, with the summary of the program modules and internal data structures given here and the comments included with the source code, it is hoped that the necessary program conversions would be relatively straightforward for programmers which understand the operations of the existing emissions processing software.

The source code for the LUMPEMIT program is organized into four files, which are listed in Table 15. These include the file LSARSPEC.FTN, which contains the specifications for the global parameters and common variables shared among the modules, which are referenced in each of the program modules using the FORTRAN INCLUDE statement. These global parameters and common variables are listed in Table 16, along with brief

Table 14. Global Parameters, Variables and Arrays Which Are Shared By the Modules of the PREPEMIT Program

Name	Type and Size[a]	Description[b]
Specifications in LUMPSPEC.FTN:		
	Miscellaneous and Program Control Variables	
LPCFIL	C*40	Name of lumping control file used
EMIFIL	C*40	Name of file used for representative emissions profile
TEMPR	R*4	Temperature in K for rate constants and alkane product stoichiometric parameters
CTEMP	R*4	Used for converting rate constant units from cm^3 $molecule^{-1}$ sec^{-1} to ppm^{-1} min^{-1}
RTEMP	R*4	RTEMP and TOV300 are used for calculating rate constant for T = TEMPR given the temperature-dependence expression (see Section III.C.4)
TOV300	R*4	
INTOH	R*4	Integrated OH level used to calculate averages for lumped groups using rate constant weighing option "1" (see Section III.D.2)
IKEY	Integer Parameter	File unit used for interactive user input
ICRT	Integer Parameter	File unit used for interactive user output
UNITn	Integer Parameters	File units for disk Input/Output (n=1-4) (File unit numbers used: 1-4 for UNIT1 through UNIT4, 5 for IKEY and ICRT)
	Specifications for Detailed Model Species (Input in Mechanistic Parameter Files)	
MAXAA	Integer Parameter	Maximum number of detailed model species (164)
MXAAPM	Integer Parameter	Maximum number of mechanistic parameters per detailed model species (32)
NAA	I*2	Number of detailed model species
NAAPRM	I*2	Number of distinct types of product yield parameters used
AANAM	C*8 (MAXAA)	Detailed model species names
AAGRP	I*2 (MAXAA)	Group type ID number for model species (see Sections III.C.3.a-b and III.D.1)
AAKOH	R*4 (MAXAA)	Detailed model species OH radical rate constant at temperature = TEMPR

(continued)

Table 14 (continued) - 2

Name	Type and Size[a]	Description[b]
AAPNAM	C*2 (MXAAPM)	Code identifying mechanistic parameter
AAPRKI	L*1 (MXAAPM)	Mechanistic parameter type: .TRUE. if this is a rate constant, .FALSE. if this is a product yield stoichiometric parameter.
AAPARM	R*4 (MAXAA,MXAAPM)	Mechanistic parameters for the detailed model species
	Specifications for Model Species Substitutions (Input Using the REPLACE Lumping Control Command)	
MAXSUB	Integer Parameter	Maximum number of substitutions which can be specified (50)
NSUB	I*2	Number of substitutions specified
SUBNAM	C*8 (MAXSUB)	Array of NSUB names to be substituted
SUBOUT	C*8 (MAXSUB)	Array of names to use to substitute those in SUBNAM.
SUBFAC	R*4 (MAXSUB)	Factor to multiply emissions amount by when making substitution
	Specifications for Explicit Model Species (Input using the SPECIES Lumping Control Command)	
MMODSP	Integer Parameter	Maximum number of explicit model species (20)
NMODSP	I*2	Number of explicit model species
MODSP	C*8 (MMODSP)	Explicit model species names
MODNC	R*4 (MMODSP)	Number of carbons for model species
MODMW	R*4 (MMODSP)	Molecular weight for model species
	Specifications for Lumped Groups (Input Using the GROUP Lumping Control Command, or Calculated by LMPSUB)	
MAXGRP	Integer Parameter	Maximum number of lumped groups (16)
NGRP	I*2	Number of lumped groups
GRPNAM	C*4 (MAXGRP)	Species name for lumped groups
GRPPNM	C*2 (MAXGRP)	Parameter prefix code for groups
GRPTYP	I*2 (MAXGRP)	Group type ID number for lumped groups
KOHMIN	R*4 (MAXGRP)	Minimum OH rate constant for groups
DRKGRP	R*4 (MAXGRP)	Default OH rate constant for groups, to use if nonzero and if rate constant weighing of emissions is used (see Section III.D.2)
GRPIND	I*2 (MAXGRP)	Emissions weighing method code for groups (see Sections III.C.a and III.D.2)

(continued)

Table 14 (continued) - 3

Name	Type and Size[a]		Description[b]
MASGRP	R*4	(MAXGRP)	Total mass emissions of species in the groups
TOTGRP	R*4	(MAXGRP)	Total molar emissions of species in the groups
RCTGRP	R*4	(MAXGRP)	Estimated amount of molar emissions into the groups which reacts with OH radicals (see Section III.D.2)
MSRGRP	R*4	(MAXGRP)	Estimated amount of mass emissions into the groups which reacts with OH radicals
RKGRP	R*4	(MAXGRP)	OH radical rate constant assigned to the groups (either weighed average or DRKGRP)
NCGRP	R*4	(MAXGRP)	Number of moles carbon in groups
NCRGRP	R*4	(MAXGRP)	Number of moles carbons in the portion of the groups which reacts with OH radicals
PRMGRP	R*4	(MXAAPM,MAXGRP)	Weighed average mechanistic parameters for the groups
	Specifications for "Non-Lumped Parameter Groups" (Emitted Detailed Model Species Not Associated with Lumped Groups)		
MAXNLS	Integer Parameter		Maximum number of non-lumped parameter species (25)
NNLS	I*2		Number of non-lumped parameter species
NLSNAM	C*8	(MAXNLS)	Species names
NLSMAS	R*4	(MAXNLS)	Mass emissions of species
NLSLVL	R*4	(MAXNLS)	Molar emissions of species
NLSNC	R*4	(MAXNLS)	Number of moles carbon in one mole of species
NROG	R*4		Molar emissions in special "unreactive" group
Specifications in SARSPEC.FTN (Data Read from SAROAD.PRM)			
MAXSPE	Integer Parameter		Maximum number of SAROAD classes (350)
MAXCLS	Integer Parameter		Maximum number of detailed model species assignments to all the SAROAD classes (500)
NSP	I*2		Number of SAROAD classes
	For Each SAROAD Class:		
SAR	I*4	(MAXSPE)	SAROAD ID number for SAROAD classes
NMIX	I*4	(MAXSPE)	Number of detailed model species assigned to each SAROAD class

(continued)

Table 14 (continued) - 4

Name	Type and Size[a]		Description[b]
IMIX	I*2	(MAXSPE)	Index number for start of list of detailed model species assigned to each SAROAD class
	For Each Detailed Model Species Assignment to a SAROAD Class:		
SPCLAS	C*8	(MAXCLS)	Detailed model species name
LVL	R*4	(MAXCLS)	Weight fraction of detailed model species for the SAROAD class (grams model species per gram SAROAD class emitted)
KPOH	R*4	(MAXCLS)	OH radical rate constant for detailed model species when representing this SAROAD class, if a value other than that read from the mechanistic parameter data set for this species is to be used.
MWI	R*4	(MAXCLS)	Molecular weight to use to calculate moles emissions of this species from mass emissions
NCI	R*4	(MAXCLS)	Number of carbons per mole for model species when used to represent this SAROAD class
Local Specifications In MAIN module for Detailed Model Species Compositions of Representative Emissions Profile			
MAXNMS	Integer Parameter		Maximum number of detailed model species in profile (200)
NMS	I*2	(MAXNMS)	Number of model species in profile
MSNAM	C*8	(MAXNMS)	Detailed model species name
MSLVL	R*4	(MAXNMS)	Molar emissions of model species
MSKOH	R*4	(MAXNMS)	OH radical rate constant of detailed model species, if different from value read for it from a mechanistic parameter file
MSNC	R*4	(MAXNMS)	Number of carbons for species
MSMW	R*4	(MAXNMS)	Molecular weight of species

[a]Species types: I*2 = INTEGER*2, R*4 = REAL*4, C*n = CHARACTER*n L*1 = LOGICAL. Size refers to maximum dimensions of arrays, which are given in parentheses if the symbol is an array.
[b]Value in parentheses is value of parameter in current version of the program. These can be changed by editing the file containing these specifications.

Table 15. Source Files and Program Modules for the LUMPEMIT Program

.FTN File	Program Module(s)	Description	Subroutines Called
LSARSPEC	-	Specifications used in all the program modules	
LUMPEMIT	MAIN	Controls overall operation of the program. Gets names of SAROAD assignment file, opens it, and calls SARINI to read it. Reads emissions data, calls LMPSAR to process it, then outputs processed data, using input and output file names specified by the user. Continuously loops back to read, process, and output more emissions data.	SARINI LMPSAR
SARINI	SARINI	Reads SAROAD assignment file and does other initializations required for processing of emissions data by LMPSAR. No user interaction.	
LMPSAR	LMPSAR	Processes a single emissions profile, using the assignments read in and initialized by SARINI. No user interaction or file input or output.	

Table 16. Global Parameters, and Variables and Arrays in COMMON Blocks Used by the LUMPEMIT Program

Name	Type and Size[a]	Description[b]
	Parameters and Numbers of Classes or Species:	
MAXSAR	Integer Parameter	Maximum number of SAROAD classes (600)
MAXASN	Integer Parameter	Maximum number of lumped model species assignments for all SAROAD classes (1000)
MAXLMP	Integer Parameter	Maximum number of lumped species used to represent emissions (60)
NSAR	I*2	Number of SAROAD classes
NLMP	I*2	Number of lumped species used to represent emissions
	For each SAROAD class for which assignments are made:	
SARNO	I*4 (MAXSAR)	SAROAD ID number for SAROAD class
ASNLOC	I*2 (MAXSAR+1)	Index number for the first lumped model species assignment for this SAROAD class. ASNLOC(NSAR+1)-1 is index no. for the last model species assignment for the last SAROAD class.
	For Each Lumped Model Species Assignment to a SAROAD Class:	
LMPNO	I*2 (MAXASN)	Index number for lumped model species
LMPFAC	R*4 (MAXASN)	Moles of lumped model species per unit mass of SAROAD species emitted. (Mass units depends on CNVFAC input into SARINI. If CNVFAC=1.0, LMPFAC is moles per gram.

[a]Species types: I*2 = INTEGER*2, R*4 = REAL*4, C*n = CHARACTER*n L*1 = LOGICAL. Size refers to maximum dimensions of arrays, which are given in parentheses if the symbol is an array.
[b]Value in parentheses is value of parameter in current version of the program. These can be changed by editing the file containing these specifications.

summaries of the purposes of each. The other three files listed in Table 15 contain each of the three major program modules of and three other files containing each of the three major program modules. The operations of the three program modules of LUMPEMIT are also indicated in Table 15, and are discussed below with regard to considerations involved in incorporation of these modules into existing emissions processing programs.

The main routine for the LUMPEMIT program is in the file LUMPEMIT.FTN. This module contains the user interface and controls all the operations of the program. It obtains the names of the SAROAD assignment file to use and opens it, and then calls the SARINI module to read its data into the common blocks specified in the LSARSPEC INCLUDE file. It then obtains the names of the emissions input data files, reads their data into memory, and then calls the LMPSAR module to calculate the lumped model species compositions which correspond to these input data. It then obtains the name of the output file(s) for the processed data and writes the data to the output file, and then loops back to request the name of the next emissions input data file to read. All user interaction and file I/O (other than the reading of the SAROAD assignment data from the already open file by SARINI) is done by this module. This module is designed primarily to provide a simple means for testing the other two modules of LUMPEMIT, which perform the major substantial operations involved in the processing of the emissions data.

The SARINI module is used to read in the lumped model species assignments of the SAROAD classes which are output by PREPEMIT, and thus serves as the interface between PREPEMIT and LUMPEMIT. This module also does other initialization operations required before the emissions data can be processed. The SARINI module is intended to do only the necessary initialization operations and thus does no operations which are dependent on the structure of the emissions data base itself (other than the requirement that the data base utilize the SAROAD classification scheme which is consistent with the input to PREPEMIT), or on the formats of the processed emissions data required by the airshed modeling program. Most of the data read in or initialized by this module are stored in the common blocks specified by LSARSPEC, which are referenced using a FORTRAN INCLUDE statement. The variables in these commons (see Table 16) consist of all of the data required by the LMPSAR module to derive the molar emissions of lumped species given the mass emissions of the SAROAD classes.

SARINI has four arguments, which are listed below in the order they appear on the argument list:

1) The INTEGER*2 number "IUNIT" is the FORTRAN unit number used to read the SAROAD lumping assignment data. This is set by the calling program and is not modified by SARINI. This file is assumed to be already open on that unit. The data on this unit is read until end-of-file, but the file is left open when the subroutine exits

2) The REAL*4 number "CNVFAC" is the units conversion factor. All the factors used to calculate molar emissions of lumped species from mass emissions of SAROAD classes are multiplied by this number. This is set by the calling program, based (for the present LUMPEMIT main routine) on the user's response to the prompt for a units conversion factor. This variable is not modified by SARINI.

3) The CHARACTER*4 array "LMPNAM" is the array into which SARINI will place the 4-character names of the lumped model species. The array of lumped model species emissions levels output by LMPSAR will always correspond to this ordering of lumped model species. This array is not included in the common specified in LASRSPEC because it is not used by LMPSAR. The calling routine allocate at least 4*MAXLMP (see Table 16) bytes for this array.

4) The INTEGER*2 variable "IRET" is used by SARINI to give the calling program a code indicating success or failure in its operations. A code of "0" indicates success; "1" indicates that there were detailed model species names in body of the assignment file which are not at the list at the beginning (see Section IV.B.3); and "2" indicates an unrecoverable error occurred when the SAROAD assignment file was read.

The module LMPSAR is called to calculate molar emissions of the lumped model species from a set of mass emissions of SAROAD classes, using the emissions assignment data produced by SARINI. This module is intended to do only the emissions conversions operations for a single emissions profile input to it through the argument list, and as such (like SARINI) its operations are intended not to depend on the organization of the emissions data base or the specific emissions input requirements of the airshed modeling program. The emissions assignment data required by

LMPSAR for its operations are in the commons specified in LSARSPEC, which is referenced using a FORTRAN INCLUDE statement. Obviously, SARINI must be called to initialize these data before LMPSAR can be called.

The input and output data for the specific emissions profile to be processed are passed through arguments to SARINI. The first three arguments give the emissions input data, which are the input to SARINI. These are:

1) The INTEGER*2 variable "NEMI," which is the number of emitted SAROAD classes in the emissions profile being processed;

2) The INTEGER*4 array "EMISAR," which gives the SAROAD ID numbers for the emissions profile; and

3) The REAL*4 array "EMIMAS," which is the mass emissions corresponding to the SAROAD classes identified in EMISAR.

These data are not modified by SARINI. The remaining three arguments are all outputs of LMPSAR, and are as follows:

4) The REAL*4 array "LMPEMI," of length NEMI (Table 16), which contains the molar emissions of the lumped model species;

5) The REAL*4 variable "TOTWT," which is the total mass emitted in the profile (i.e., the sum of the data in EMIMAS); and

6) The REAL*4 variable "UNKWT," which is the total mass in the emitted profile for which no assignments of lumped model species could be made.

Note that the ordering of the data in the LMPEMI array is based on the same ordering of lumped model species as given in the LMPNAM array of model species names output by the SARINI module. Thus, it is not necessary for LMPSAR to also output the lumped model species names which correspond to the assignments of their emissions, since these will never vary within a given LUMPEMIT run. This ordering of lumped model species is also known to the airshed modeling program, since PREPEMIT includes this information in a lumped species parameter file which is intended to be read by the airshed modeling program during the mechanism initialization process (as described in Sections IV.C.3). Thus, single arrays of molar emissions rates, as output by LMPSAR in the LMPEMI arrays, is all that is required by the airshed modeling program to characterize an emissions profile.

The organization of LUMPEMIT is such that most, if not all, modifications to the program to make it compatible with existing emissions data bases or airshed modeling programs would be made in the main module in the PREPEMIT.FTN file. Alternatively, the SARINI and LMPSAR modules could be incorporated into existing emissions processing programs. The SARINI and CMPSAR subroutines contain the major code required for processing of emissions data for input into airshed models based on methods and procedures discussed in Section III, and these modules are designed so they should not have to be extensively modified for them to be useful in specific airshed modeling environments.

Note, however, that if large amounts of emissions data are to be processed at the same time, it may be advisible to re-code LMPSAR so it will process many arrays of emissions inputs at once. In particular, significant savings in computer time could be attained if the SAROAD numbers associated with the input emissions arrays are the same for all profiles, and are the same as used in the assignment arrays output by SARINI. In addition, the present coding of LMPSAR does not take advantage of capabilities of vector processing machines, which could allow for significant savings in computing time when many arrays of emissions are processed at once. The logic of LMPSAR is sufficiently simple that the appropriate modifications in this regard should be relatively straightforward.

3. Formats of PREPEMIT and LUMPEMIT Output Files

In this section, the data formats of the output files produced by PREPEMIT and LUMPEMIT are discussed. These files are not normally intended to be maintained by the user, but are used as input into another program. These files are (a) emissions parameter files output by PREPEMIT; (b) SAROAD assignments files, also output by PREPEMIT, and (c) processed emissions files output by LUMPEMIT.

a. Emissions Parameter Files

These files, also called "lumped species parameter files" are ouput by PREPEMIT for the purpose of passing to the airshed modeling program the list of lumped model species and their parameters. This file is produced by the LMPSUB module of PREPEMIT and is read by the RDPRM example subroutine to the airshed model program. These files can contain four different types of records, as indicated in Table 17. These records

Table 17. Format of Data in Emissions Parameter Files Output by PREPEMIT

Column	FORTRAN Format Code	Description of Data
1a. Lumped Group Rate Constant Input		
1-2	"K("	Code identifying this as a rate constant input record
3-4	A2	Parameter prefix code used for all parameters for this lumped group
4-6	A2	Code for type of reaction: "OH" = OH radical reaction; "O3" for ozone reaction; "OA" for $O(^3P)$ reaction; and "N3" for NO_3 radical reaction
7-8	") "	Code identifying this as a rate constant input record
9-20	1PE12.4	Rate constant (ppm^{-1} min^{-1} units)
1b. Lumped Group Mechanistic Parameter Input		
1-2	A2	Parameter prefix code used for all parameters for this lumped group
3-4	A2	Parameter suffix code for this type of mechanistic parameter
5-8	4X	(Must be blank)
9-20	1PE12.4	Parameter value
2. Emitted Lumped Model Species Specification Input. (Order these records are input determine order of emitted lumped species which is output by LUMPEMIT.)		
1-4	"RHC("	Code used to indicate that this is an emitted model species specification input record
5-8	A4	Lumped model species name
9	")"	Part of code identifying this type of input
10-15	F6.2	Number of carbons for this model species
16-24	F9.2	Molecular weight for this model species
3. Relative or Absolute Molar Emissions of Lumped Species in the Representative Emissions Profile Input into PREPEMIT. (These records are ignored by LUMPEMIT.)		
1-4	A4	Lumped model species name
5-8	4X	(Must be blank)
9-20	1PE12.4	Molar emissions amount

(continued)

Table 17 (continued) - 2

Column	FORTRAN Format Code	Description of Data
4.	Comments.	(Ignored by RDPRM)[a]
1	"!"	Code identifying this record as a comment
2-80	79X	Comments

[a]PREPEMIT outputs comments at the beginning of the file indicating the lumping control file and representative emissions profile file used as input. It also outputs comments for each lumped group indicating the emissions weighing method employed, the moles and moles carbon emitted, the average number of carbons per mole and molecular weight and (for groups using rate constant weighing) the estimated amount reacting with OH radicals and their average number of carbons and molecular weights.

can consist of (1) the kinetic or mechanistic parameters for the "lumped groups," i.e., the rate constants or product yield stoichiometric parameters for the model species whose parameters depend on the set of emitted species they represent; (2) the name of a model species can be used to represent organic emissions, together with its molecular weight and number of carbons; (3) the molar emissions amounts of the model species which correspond to those in the representative emissions profile input into PREPEMIT, or (4) comments indicating the name of the lumping control or emissions data file input into PREPEMIT, or giving other information which is ignored by the RDPRM subroutine. If there are no lumped groups in the mechanism, no records of type (1) are in this file. There is a record of type (2) for each model species (both explicit and lumped groups) in the mechanism which can be used to represent emitted organics. Records of type (3) are ignored by RDPRM. They are included in the parameter file only if the user of PREPEMIT specified that these data were to be included in this file; if not, PREPEMIT outputs these data into a separate file. Comment records those which begin by a "!" character, as is the case for many other types of files.

b. Lumped SAROAD Assignments Files

The PREPEMIT program also outputs lumped SAROAD assignment files, which are intended to be read by LUMPEMIT or the equivalent program which processes emissions input data for the airshed model. These files are written by the SARSUB module of PREPEMIT, and are read by the SARINI module of the emissions processing program. These files contain all the data the emissions input processing program needs to convert mass emissions of SAROAD classes into molar emissions of species in the airshed model's chemical mechanism. This includes, for each SAROAD class, (1) the SAROAD ID number; (2) the number of lumped model species used; and (3) the number of moles of model species per gram of emissions of the SAROAD class. This file also contains additional data which is generated by PREPEMIT in the process of deriving these assignments, which are output for diagnostic purposes, but which are ignored by the LUMPEMIT program. The format of the data in these files, including the additional diagnostic data which are not used by LUMPEMIT, are indicated in Table 18.

c. Processed Emissions Data Files

Emissions data processed in a form which should be usable by the airshed modeling program are output by LUMPEMIT in the processed emissions data files. As discussed in Section III.B.2, these files can have two formats, depending on the option selected by the user when LUMPEMIT is run. The format which will probably be the most useful in multi-cell airshed model applications is the format produced when the user selects to have the processed emissions data all together in a single file. Emissions data files with this format simply consist of arrays of molar emissions data files. These files simply consist of arrays of molar emissions of the lumped model species, with the emissions for the species given in the order they appear at the list in the initial records of the SAROAD assignment file (see Table 18), and in the "RHC(name)" records in the emissions parameter file (see Table 17). If NLMP is the number of lumped model species listed at the beginning of the SAROAD assignment file, then the processed emissions file has a group of NLMP real numbers, output in (1P6E12.4) format, for each emissions profile input into LUMPEMIT. The airshed modeling program can determine which model species each of these emissions amounts correspond to because the number and

Table 18. Format of Data in Lumped SAROAD Assignments Files Output by PREPEMIT

Column	FORTRAN Format Code	Description of Data
Group 1:	First Record in File.	
1-4	I4	Number of lumped model species (NLMP)
5-40	1X,A35	Name of lumping control file used when this data set was created
40-76	1X,A35	Name of file used for the representative emissions profile when this data set was created.
Group 2:	One record for each lumped model species. (NLMP records)	
1-4	A4	Species name
5-12	F8.3	Number of carbons
13-22	F10.3	Molecular weight
Group 3:	Remaining Records in File. One set of records for each SAROAD class for which lumped model species assignments	
1-5	I5	SAROAD ID number
6-8	I3	Number of lumped model species assigned to this SAROAD class
9	1X	(Should be blank)
10-13	A4	Name for first lumped model species assigned to this SAROAD class
14-24	1PE11.3	Moles first model species per gram SAROAD species emitted
25-31	F7.2	Number of carbons for this model species[a]
32-39	F8.2	Molecular weight for this model species[a]
40-50	1PE11.3	OH radical rate constant for this model species, if it is a lumped group[a]
		Continuation records for SAROAD classes which are represented by more than one lumped model species
1-9	9X	(Must be blank)
10-13	A4	Name for mext lumped model species assigned to this SAROAD class

(continued)

Table 18 (continued) - 2

Column	FORTRAN Format Code	Description of Data
14-24	1PE11.3	Moles of this model species per gram SAROAD species emitted
25-31	F7.2	Number of carbons for this model species[a]
32-39	F8.2	Molecular weight for this model species[a]
40-50	1PE11.3	OH radical rate constant for this model species, if it is a lumped group[a]

[a]These data are output for diagnostic purposes only, and are ignored by LUMPEMIT. They refer to the number of carbons, molecular weights and (if applicable) OH radical rate constant that would be assigned to this lumped model species if this SAROAD class were the only species emitted. These in general would be different to the values assigned for the lumped model species in the lumped parameter output file, which are appropriate for the representative emissions profile as a whole.

ordering of the model species is the same as that given in the emissions parameter file, which (as discussed in Section IV.C.3) is input into the airshed modeling program as part of the mechanism initialization process.

If the user of LUMPEMIT selects the option of having the processed emissions data output in separate files for each emissions profile, the output files have a different format. In this case, the emissions data consist of one record for each lumped model species, giving the name of the species and the molar emissions amount [in (A4,1PE16.4) format] for the species. The initial records of the file consist of comments (indicated by a "!" as the first character of the records) giving the names of the lumped SAROAD assignment file and the emissions file which were used to create this file, and giving a summary of the total masses of emissions and unassigned emissions in the emissions input file. The example emissions output files listed in Appendix A and discussed in Section V are in this format.

C. Example Subroutines for Implementation of Prepared Mechanism into Airshed Model Software

The output of the model preparation and emissions processing programs are designed with no specific airshed modeling software system in mind, and some modifications of airshed model software will be required to utilize the mechanism processing capabilities of this program. This modification should be relatively straightforward if the airshed model software is designed in a modular manner and its internal operations with regard to calculations of effects of chemical transformations are reasonably well documented and understood. To assist in this conversion (and also to test and de-bug this version of the preparation program), we have written four FORTRAN subroutines which can serve as examples of chemical transformation subroutines to an airshed modeling program which implement mechanisms processed by the preparation program, and emissions data processed using the PREPEMIT and LUMPEMIT programs. These example subroutines, and the data structures they employ, are discussed in this section.

The four example airshed model subroutines whose source code is included with this version of the preparation program are called RDMOD, RDPRM, NEWRK, and NEWPHK. Briefly, RDMOD is used to initialize the parameters for the chemical mechanism, most of which are read from the .MOD file produced by the preparation program; RDPRM is called by RDMOD as part of the mechanism initialization to read the parameters output by PREPEMIT for the model species used for emitted organics; NEWRK is used during the model simulation to update temperature-dependent rate constants; and NEWPHK is used during the model simulation to update light-dependent rate constants. The source code for these four subroutines are given in .FTN files of the same name, and a fifth file, MODLSPC.FTN, contain the specifications for the parameters, and COMMON variables and arrays which are used to contain the data for the chemical mechanism. This MODLSPC file is INCLUDE'd in the source for the RDMOD, RDPRM, NEWRK, and NEWPHK subroutines, and should also be INCLUDE'd by other modules of the airshed modeling program which may need reference to these data, such as input and output subroutines which may need to reference the species names, etc. The parameters and COMMON data specified in the MODLSPC.FTN file, and the use and arguments of the three example airshed model subroutines are discussed in the following sections.

1. Mechanism Specific Parameters and COMMON Data Specified in the MODLSPC.FTN Specification File

The parameters and COMMON variables and arrays whose specifications are given in the MODLSPC.FTN file consist of all of the data needed to implement and document the chemical mechanism in the airshed modeling program. These are initialized by RDMOD and the modules it calls (i.e., RDPRM to initialize emissions related parameters and the PHKINI entry of NEWPHK to initialize parameters used to calculate photolysis rates), and are utilized by the NEWRK and NEWPHK routines when they update the temperature- and light-dependent rate constants. The variables and parameters specified in the MODLSPC.FTN file, are listed and briefly described in Table 19, along with an indication of which of the example subroutines (other than RDMOD) utilize or modify them.

2. The RDMOD Subroutine for Initialization and Documentation of the Chemical Mechanism

The RDMOD subroutine is called once during the initialization phase of the airshed modeling program to read in and initialize the mechanism specific data from the mechanism's .MOD file and optionally to write onto an output file a listing of species, reactions, and parameters in the chemical mechanism. The data read from the .MOD file are placed in the variables and arrrays in the COMMON blocks specified in the MODLSPC.FTN INCLUDE file, where they are available for use by the other modules of the airshed modeling program, including RDPRM, NEWRK and NEWPHK. In addition to reading in the data from the .MOD file, the RDMOD subroutine calls NEWRK to calculate the rate constants for the temperature specified as an argument to RDMOD, and calls PHKINI (a module in NEWPHK) to do the initialization required for calculating photolysis rate constants. RDMOD will also optionally call RDPRM to read data used for emissions processing which are output by the PREPEMIT program. If the mechanism listing option is selected, the photolysis rate constants are initialized to their values for solar zenith angle of zero (these being output in the mechanism listing for documentation purposes), otherwise they are undefined. (It is assumed that the airshed model software will cause the appropriate initial values for the photolysis rates to be calculated before the simulation begins, so this is not done by RDMOD.) Whether RDMOD produces mechanism listing output is determined by a logical variable in the argument list.

Table 19. Parameters and Variables and Arrays in COMMON Blocks Used in the Example Airshed Model Subroutines for Implementing Chemical Mechanisms Produced by the Preparation Program

Name	Type and Size[a]	Description[b]
	Specifications for Species	
MAXNS	Integer Parameter	Maximum number of species (140)
MAXACT	Integer Parameter	Maximum number of integrated (active) species (60)
MAXBO	Integer Parameter	Maximum number of buildup only species (30)
MAXCON	Integer Parameter	Maximum number of constant species (50)
MAXDUM	Integer Parameter	Maximum number of dummy species (10)
NS	I*2	Number of species
NACT	I*2	Number of active species
NSC	I*2	Number of constant species
NSB	I*2	Number of buildup species
NDUM	I*2	Number of dummy species
NAME	C*16 (MAXNS)	Species names. The first NSC names are for the constant species, the next NSB are buildup-only, the next NACT are active, and the remaining NS-NSB-NACT are steady-state species names. The ordering of species within each type is the same as the ordering required in the concentration arrays used as arguments to the mechanism specific subroutines.
CONCO	R*4 (MAXNS)	Default initial concentrations for the constant, buildup-only, dummy, and active species, respectively. (Same species ordering as for NAME.) Not defined for steady state species.
MWT	R*4 (MAXNS)	Default molecular weights for the non-steady state species. Same ordering as above.
CNO	R*4 (MAXNS)	Default number of carbons for the non-steady state species. Same ordering as above.
NNO	I*2 (MAXNS)	Default number of carbons for the non-steady state species. Same ordering as above.
	Specifications for Emitted Model Species for Organics (Read by RDPRM for use by emissions input modules)	
MAXRHC	Integer Parameter	Maximum number of emitted organic species (60)
NRHC	I*2	Number of emitted organic species
INDRHC	I*2 (MAXRHC)	Species index number for emitted organics

(continued)

Table 19 (continued) - 2

Name	Type and Size[a]	Description[b]
	Specification for Reactions and Kinetic Parameters	
MAXRXN	Integer Parameter	Maximum number of reactions (220)
MAXKBF	Integer Parameter	Maximum number of kinetic parameters (3*MAXRXN)
NRXN	I*2	Number of Reactions (Used by NEWRK)
LENKBF	I*2	Number of kinetic parameters in the KPBUF array
RXNLBL	C*4 (MAXRXN)	Reaction Label (Used by RDPRM)
RK	R*4 (MAXRXN)	Rate constant arrays. (Initialized by RDMOD. May be modified by RDPRM. Updated by NEWRK and NEWPHK. Use for the "RK" argument to the mechanism specific subroutines)
RXTYP	I*2 (MAXRXN)	Type of reaction kinetics: 0 = SAMEK type; 3 = CONST type; 4 = extended Arrhenius thermal; 5 = FALLOFF thermal; 7 = photolysis reaction. (Used by NEWRK)
LKBUF	I*2 (MAXRXN)	Use depends on type of reaction kinetics. For types 3, 4, and 5, gives index for first kinetic parameter in KPBUF array. For type 0, gives reaction number whose rate constant is used. For type 7, gives number of photolysis set used. (Used by NEWRK)
KPBUF	R*4 (MAXKBF)	Kinetic parameters for non-photolysis reactions. Number and use of parameters depends on type of kinetics, as follows: type (3): rate constant; type (4): A, Ea, and B; type (5): A0, E0, B0, AI, EI, BI, F, and N. First parameter for given reaction pointed to by LKBUF array. Not used for type 0 and 7 reactions. (Used by NEWRK)
	Specifications for Coefficients	
MAXCO	Integer Parameter	Maximum number of coefficients (470)
MAXCOV	Integer Parameter	Maximum number of variable coefficients (420)

(continued)

Table 19 (continued) - 3

Name	Type and Size[a]	Description[b]
NCOCO1	Integer Parameter	Index number for the first constant coefficient. (Always MAXCOV+1)
NCOEFV	I*2	Number of variable coefficients
NCOC1	I*2	Index number for first constant coefficient in the COEF array
NCOC2	I*2	Index number for the last constant coefficient in the COEF array
COEFNM	C*8 (MAXCOV)	Variable coefficient names. (Used by RDPRM)
COEF	R*4 (MAXCO)	Coefficient values. First NCOEFV are values of the variable coefficients, NCOC1 to NCOC2 values are for the constant coefficients, and should not be modified. (Initialized by RDMOD. May be updated by RDPRM. Use for "CO" argument to the mechanism specific subroutines)
	Specification of Photolysis Data and Parameters (All used by NEWPHK unless indicated otherwise)	
MAXPHK	Integer Parameter	Maximum number of photolysis sets (20)
NSOLWL	Integer Parameter	Number of wavelengths in photolysis sets after being processed for calculation of solar photolysis rates. This is the number of absorption coefficient x quantum yield values which are saved in the .MOD file. (always 27)
NPHOTK	I*2	Number of photolysis sets.
PHOTNM	C*8 (MAXPHK)	Photolysis set name or .PHF filename. (Not used by NEWPHK - can be made local to RDMOD if desired.)
IPHRK	I*2 (MAXPHK)	Reaction number for the first photolysis reaction of each type. This gives the location in the RK array where the DIFUN subroutine requires the rate constant calculated for the photolysis set to be placed.
ABSQY	R*4 (NSOLWL,MAXPHK)	Absorption coefficient and quantum yield products for each of the photolysis reactions for the standard set of NSOLWL wavelengths.

(continued)

Table 19 (continued) - 4

Name	Type and Size[a]	Description[b]
DPHOTK	R*4 (MAXPHK)	The photolysis rates for each photolysis calculated for the zenith whose cos = CZLOW. Initialized by RDMOD and used by NEWPHK for calculating photolysis rates at high zenith anbles.[c]
	Mechanism Independent Parameters and Variables Required by NEWRK or NEWPHK for Calculation of Rate Constants.	
	1. Constant Variables or Parameters	
TREF	R*4	Reference temperature used by NEWRK for calculating temperature dependent rate constants. See Section II.B.2 or II.B.5.a. Read from the .MOD file by RDMOD, and must not be modified.
CZLOW	Real Parameter	Cosine of the highest zenith angle for which the equation used in NEWPHK to calculate solar irradiances can be applied. Light intensities assumed to be proportional to zenith angle for angles whose cosine are greater than CZLOW. See source listing of NEWPHK for details. (7.0×10^{-2})
	2. Problem-Specific Variables. (NOTE: These variables must be set by other modules in the airshed modeling program PRIOR to calling RDMOD to initialize the chemistry. The example subroutines assume these are already defined.)	
LAT	R*4	Latitude, in degrees. Required by RDMOD for initialization of parameters used by NEWPHK.
DEC	R*4	Solar declination angle, in degrees. Required by RDMOD for initialization of parameters used by NEWPHK.
TSTART	R*4	Difference between clock time used by model and true solar time, in minutes. (i.e., TSTART is around -60 if daylight time is used.) Used by NEWPHK.
CZMIN	R*4	Cosine of the effective "zenith angle" to be used in calculating photolysis rates at night. If zero, nighttime photolysis

(continued)

Table 19 (continued) - 5

Name	Type and Size[a]	Description[b]
		rates are zero. Nonzero values may result in better performance for some solvers.
	3. Work Variables Used by NEWPHK (c)	
SINLD	R*4	Sin(LAT) x Sin(DEC)
COSLD	R*4	Cos(LAT) x Cos(DEC)

[a]Species types: I*2 = INTEGER*2, R*4 = REAL*4, C*n = CHARACTER*n L*1 = LOGICAL. Size refers to maximum dimensions of arrays, which are given in parentheses if the symbol is an array.
[b]Value in parentheses is value of parameter in current version of the program. These can be changed by editing MODLSPC.FTN.
[c]These are set by RDMOD calling the PHKINI initialization entry in NEWPHK, and are then used by NEWPHK for calculating photolysis rates. These could be local to the NEWPHK routines, and are only included in the COMMON's here to emphasize that they must be preserved throughout the simulation, and must not be over-written if NEWPHK is overlayed.

The RDMOD subroutine should be called before the initial reactant concentrations and emissions data are input, since the names of the species in the mechanism species used for organic emissions are not known to the program until RDMOD reads them from the .MOD file, and the species which are used to represent emissions are not known until RDPRM (which is called by RDMOD) reads them from the emissions parameter file. On the other hand, the variables LAT, DEC, and CZMIN (see Table 19) should be defined before this subroutine is called, and an output listing file must already be open, even if no mechanism documentation output is desired. The input (.MOD) file used by RDMOD to read the mechanism parameters, and the mechanistic parameter file used by RDPRM (if called) must also both be open when RDMOD is called.

RDMOD has seven arguments, listed below in the order they appear on the argument list:

1) A CHARACTER*16 variable MOD, which gives the name of the mechanism. It is used only in the mechanism listing documentations, if it is produced.

2) An integer variable IUNIT is used as the FORTRAN unit number for the .MOD file which is used for input. The appropriate .MOD file is assumed to be already open on that unit, but that no READ's from that file have been made. RDMOD does not close this file, but it can be closed by the calling program after RDMOD is completed, since all useful data will have been read, unless the .MOD file is included as part of the overall input stream. The only input RDMOD requests are .MOD file data read from this unit.

3) An integer variable OUT is used as the FORTRAN unit number for the listing output unit. This is the unit that RDMOD uses to output the listing of the chemical mechanism (if requested), and to output error messages. RDMOD assumes that that output unit is already open, and that it can accept up to 132 character output records, with the first character of each output record being a FORTRAN carriage control character. This is the only output unit used by RDMOD unless the program has to be aborted due to an error.

4) A logical variable DOCOUT is used to tell RDMOD whether to produce a listing of the mechanism on the output unit OUT. If DOCOUT is .TRUE., a listing is produced, otherwise not.

5) A real variable TEMP gives the temperature (in degrees K) which is used in the initial calculation of the thermal rate constants. If TEMP is set to zero when RDMOD is called, the default temperature of the mechanism is used, and TEMP is set to that default temperature. Otherwise, the default temperature of the mechanism is ignored.

6) An integer variable IRXP is used to tell RDMOD whether to call RDPRM to read parameters for emitted organics which are output by PREPEMIT. If IRXP is zero, RDPRM is not called. If IRXP is nonzero, it is the FORTRAN unit number RDPRM should use to read the emissions parameter data. The emissions parameter file is assumed to be open on this unit. Note that IRXP cannot be the same as IUNIT, since RDPRM is called to read the emissions parameter data (on unit IRXP) before all the .MOD file data (on unit IUNIT) has been read.

7) IERR is an integer variable which is used by RDMOD to tell the calling program whether it has been successful. If IRET is zero, then RDMOD has been executed successfully, and the airshed model calculation

can proceed. If it is positive, then an error occurred in RDMOD, and if it is negative, an error occurred when RDMOD called RDPRM.

Note that of these arguments, MOD, IUNIT, OUT, IRXP and (if nonzero) TEMP are used as input to RDMOD and are not modified. If TEMP is zero, it is set to the mechanism default. The main output of RDMOD are the variables and arrays in common blocks specified by MODLSPC, which are initialized by this subroutine. The only COMMON variables which RDMOD (or routines called by it) does not initialize (other than the photolysis rates, as indicated above) are LAT, DEC, CZMIN, and TSTART. The first three of these must be defined before RDMOD is called; the latter is not required until the initial values for the photolysis rates have to be calculated.

3. The RDPRM Subroutine for Initialization of Emissions Parameters

If the emissions processing system discussed in Section III is employed in the airshed model, then the RDPRM subroutine should be called during mechanism initialization process. This subroutine reads the lumped species parameter files which are output by the PREPEMIT program, derived from the composition of the representative emissions profile, and specifications of lumping methods and the detailed assignments for emitted species which are input into that program. The specific data it reads are as follows: (1) the rate constants and product yield coefficient values for model species whose parameter values depend on the set of emitted species which they represent (i.e., for the "lumped groups"); (2) the list of all species used in the mechanism to represent emitted organics, given in the order they are output by the LUMPEMIT emissions pre-processing program; and (3) the molecular weights and number of carbons used for all the species used to represent emited organics. RDPRM expects these data to be in the format produced by PREPEMIT for lumped species (or emissions) parameter files, which is given in Table 17 above.

The recommended procedure for calling RDPRM is for the airshed modeling program to have it called by RDMOD. This will be done if the airshed modeling program module which calls RDMOD to open the emissions parameter file on some input unit different from that used for the .MOD file (which it also has to open before RDMOD is called), and pass the FORTRAN unit number used for the emissions parameter file to RDMOD as the 6'th (IRXP) argument. Alternatively, the airshed modeling program could call RDPRM after RDMOD is called; the main difference is that the parameters input or updated by RDPRM will not be included in the output file produced by RDMOD to document the mechanism. RDPRM should not be called before RDMOD is called, since RDPRM needs to know the parameter symbols used in the mechanism which are read by RDMOD from the .MOD file, and also RDMOD would over-write any data input by RDPRM.

RDPRM expects the data it reads from the emissions parameter file to be consistent with the chemical mechanism data read from the .MOD file, or it will abort and return to the calling program with an error code. In particular, except for the special species "NROG," RDPRM expects all of the four-character codes used by emitted model species in the emissions parameter file to be on the list of active or dummy species names for the mechanism. The special species "NROG" is used to represent non-reacting carbon, which is not required to be on the list of species in the mechanism. (Note that because the names for emitted species processed by the emissions programs can have a maximum of four characters, only four characters can be used in the mechanism for species which may be emitted, even though the preparation program allows for up to 16 characters for species names.) RDPRM also expects that the codes used in the emissions parameter file for rate constants or product yield parameters will be found on the lists of rate constant labels or coefficient names, respectively, in the mechanism. The only exception to this is that the program will not treat it as an error if it does not find a coefficient name of the form "xxNC" in the mechanism. Parameters with the suffix "NC" have the reserved meaning "number of carbons," and such parameters may not necessarily be used by the mechanism, though they are automatically generated by PREPEMIT for all lumped groups.

It should be noted that in addition to updating the rate constant and parameter arrays for emitted species based on the output of a PREPEMIT run, RDMOD will also create a list of the model species which are used to represent reactive organic (ROG) emissions. This list is created in the form of species index numbers in the INDRHC array, and the variable NRHC is set to the number of such species (see Table 19). Each species for which an "RHC(" input record is included in the emissions parameter file (see Table 17, above) is included on this list, in the order they are input. Note that this list includes both explicit model species and lumped groups. This INDRHC array is useful because if LUMPEMIT (or an equivalent emissions processing program) uses the SAROAD assignment file output from the same run of PREPEMIT as used to produce the emisions parameter file read by RDPRM, then the ordering of the emitted lumped species LUMPEMIT outputs will always correspond to the ordering of species in the INDRHC array produced by RDPRM. If this is the case, the emissions input module of the airshed modeling program can then use the index numbers in this array to allocate emissions input into the airshed model to the appropriate species in the mechanism, without having to refer to, or search through, the names of the species in the mechanism.

The RDPRM subroutine has three arguments, all integers. These are listed below in the order they appear on the argument list.

1) IUNIT is the FORTRAN unit number for emissions parameter file. The emissions parameter file must already be open on this unit when RDPRM is called. RDPRM reads from this file until end-of-file, or until an "END" record is input.

2) OUT is the FORTRAN unit number to use to output error messages. It must be open for output. However, it is only used if an error occurs.

3) IERR is used for RDPRM to inform the calling program whether it is executed successfully, and whether it terminated because of an end-of-file on the input unit, or because an "END" record was input. If RDPRM encounters no errors, IERR is set to "0" if it read the input file until end-of-file, or IERR is set to "1" if it terminated reading because of an "END" record being input. A negative value of IERR is returned if an error occurred. IERR is set to "-1" if it input a record it could not process, or which has a parameter or species name which is not valid for this mechanism. (The offending record is output on the unit OUT.) IERR is set to "-2" if RDPRM is called with a zero or negative value of IUNIT.

4. <u>The NEWRK Subroutine for Calculation of Temperature Dependent Rate Constants</u>

The NEWRK subroutine should be called during the model simulation at any time the temperature changes. It is also called by RDMOD to calculate the rate constants to the mechanism default of the specified initial temperature. If the model simulation is carried out at constant temperature, it need not be called again. If the temperature varies from cell to cell in the airshed model, NEWRK should be called whenever the simulation changes the cell for which the chemistry is being processed. If it is called, it must be called before the mechanism specific subroutine DIFUN is called. This subroutine does no input or output.

The only argument to NEWRK is a real variable which contains the temperature, in degrees K, for which the rate constants are to be calculated. The results are placed in the RK array in the one of the COMMON blocks defined in MODLSPC.FTN. The data in the COMMON arrays temperature for each cell, before the mechanism specific subroutine DIFUN is called. The COMMON data required by NEWRK the constants NRXN and TREF, and the arrays RXTYP, LKBUF, and KPBUF.

5. The NEWPHK Subroutines for Calculation of Photolysis Rate Constants

All the codes required to calculate rate constants for the photolysis reactions are in the file NEWPHK.FTN. This file actually contains three subroutines: (1) the subroutine PHKINI, which is called by RDMOD to initialize the data used to calculate the photolysis rates; (2) the subroutine NEWPHK which is used to calculate the photolysis rate constants for a given time of day; and (3) the subroutine SOLPHK, which is called by both PHKINI and NEWPHK to calculate the photolysis rate constants given a solar zenith angle. This SOLPHK subroutine is not normally called directly by routines outside this file. The operations and arguments to these three subroutines are discussed below.

The initialization subroutine PHKINI is called by RDMOD to define the variables SINLD and COSLD, which depend only on the latitude and the declination, and the values in the array DPHOTK, which depends on CZLOW and the absorption coefficients and quantum yields. SINLD and COSLD are used by NEWPHK to calculate the solar zenith angle given the time of day, and the values in DPHOTK are used to calculate photolysis rates when the cosine of the zenith angle is less than CZLOW. (See Table 19 and the NEWPHK.FTN source file for more details concerning these variables.) Its only argument is a logical variable which, if .TRUE., tells NEWPHK to also calculate the photolysis rates for a solar zenith angle of zero, which can be output for documentation purposes. Otherwise, the photolysis rate constants are set to values corresponding to a solar zenith angle whose cosine is CZLOW, or approximately 78 degrees (this being a result of the calculation of DPHOTK). This subroutine need be called only once, unless there is a change in the mechanism, latitude, or declination in the airshed model calculation. It requires the COMMON variables and LAT and DEC, the parameter CZLOW, and the contents of the COMMON array ABSQY to be defined.

The subroutine NEWPHK is called to calculate the photolysis rate constants at appropriate intervals throughout the airshed model calculation. It should also be called at the start of the simulation to initialize the photolysis rates for the time of day the simulation begins (even if the simulation begins at night), since this function is not performed by RDMOD. This subroutine has two arguments, TIME and FAC. TIME is a real number which gives the time of day in minutes after midnight for which the rate constants are to be calculated. FAC is a real number giving an overall light intensity factor, which is multiplied by the theoretically calculated photolysis rates to simulate less-than-clear-sky conditions. If clear sky rate constants are desired, a value of 1.0 should be used. A value of zero is legal, and results in all the photolysis rates being zero (regardless of the value of CZMIN). NEWPHK uses the value of TIME and the COMMON variables COSLD, SINLD, and TSTART to calculate COSZ, the cosine of the solar zenith angle. If COSZ is less than CZLOW, it calculates the photolysis rates as FAC*(maximum of COSZ and CZMIN)/CZLOW times the values in DPHOTK (the photolysis rates calculated by SOLPHK when COSZ is CZLOW); otherwise, it calls SOLPHK to calculate the photolysis rates.

The subroutine SOLPHK is used to calculate the photolysis rates if the zenith angle is less than approximately 86 degrees (as determined by the value of the parameter CZLOW). Its arguments are COSZ, the cosine of the zenith angle, and FAC, the overall light intensity factor. It is called by PHKINI with COSZ set to CZLOW for the purpose of defining the DPHOTK array, and by NEWPHK when COSZ is greater than CZLOW. The method of calculation of the photolysis rates are the same as employed in other SAPRC modeling software, which is described in detail elsewhere (Carter et al. 1986), and in the SOLPHK source code. Briefly, they are calculated using the solar irradiances using parameters empirically fit to the actinic irradiances given by Peterson (1976), using his "best estimate" surface albedos, and the absorption coefficient and quantum yield products given in the ABSQY array for the various photolysis processes. The empirical fits to Peterson's actinic irradiances are only valid for solar zenith angles of 86 degrees or less, so the value of CZLOW is thus set accordingly. A standard set of wavelengths is used, which is given in the source file for SOLPHK as a dummy DATA statement for the pseudo-variable SOLWL, and this must correspond exactly to the SOLWL array which is local to the SAVERX subroutine of the preparation program. It is important to recognize that if there are any changes to the ways the photolysis rates are calculated, the code involved in the processing of the absorption coefficient and quantum yield data in the preparapation program subroutine SAVERX may also need to be changed.

Both NEWPHK and SOLPHK place the values of the photolysis rates they calculate in the RK array, which is in a COMMON as defined by MODLSPC.FTN. The COMMON array IPHRK is used to tell these subroutines the location in the RK array to place the photolysis rates calculated for each photolysis set. Note that if more than one reaction uses the same photolysis set, only the location in the RK array for the first such reaction need be defined, since the mechanism dependent code in DIFUN uses this location to obtain the rate constants for all the reactions using this photolysis set.

V. DATA FILES AND EXAMPLE CALCULATIONS FOR IMPLEMENTATION

The distribution of the software discussed in this document include data files implementing the gas-phase transformation mechanisms and emissions processing assignments which have been developed for implementation with this software. These include: (1) preparation program input files for preparation of these mechanisms for use in airshed models; (2) input files for the emissions processing programs giving the emissions assignments which have been made for processing emissions data for use with these mechanisms; and (3) example input and output data for test calculations utilizing this software and these data files. These data files and test calculations illustrate the use of this software with this chemical mechanism and emissions assignments, and can be utilized to test for successful implementation of this software and these mechanisms on other computer systems.

A. Chemical Mechanisms Implemented in the Data Files

The data files which implement the chemical mechanism which have been developed for use with this software are listed in Table 20. These mechanisms are described and documented elsewhere (Carter, 1988), and only their major features are summarized here. Briefly, a series of chemical mechanisms have been developed which employ varying degrees of chemical detail or condensation in representing the gas-phase reactions of the emitted species and their reactive photooxidation products. These are all consistent in their common reactions, which are based primarily on previous mechanisms developed at SAPRC (Carter et al., 1986, 1987; Lurmann et al., 1987) updated based on the results of recent evaluations of atmospheric chemistry [primarily NASA (1987) for the inorganic reactions and Atkinson (1988) for the organics], and adapted for use with the emissions lumping system described in Section III. The appropriate mechanism to utilize in a particular application depends on the level of detail required for the application, and on considerations of computer time or resources available for the application. For example, in multi-celled airshed model calculations which require extensive computer resources and time, a high degree of condensation of the chemical mechanism is usually necessary, with only a limited number of species available to represent

Table 20. Data Files Implementing the Chemical Mechanisms which Were Developed for Use with this Software

File Names			Description
Detailed Mechanism	Lumping Level #1	Lumping Level #2	
			Preparation Program Input files.
LMPAE221.PRP	LMPCE221.PRP	LMPDE221.PRP	Main Preparation program input file for two lumped alkanes, two aromatics, ethene and one higher alkene. Reactions of H2O2 and -OOH are included.
(a)	(a)	LMPBE221.PRP	As above, but with reactions of H2O2 and -OOH removed.
LMPAE433.PRP	(a)	(a)	Main Preparation program input file for four lumped alkanes, four aromatics, ethene and three higher alkenes. H2O2 and -OOH reactions included.
NOXAIR.RXN	NOXAIR.RXN	NOXAIR.RXN	Inorganic reactions
SO2.RXN	SO2.RXN	SO2.RXN	Reactions of SO2
RO2.RXN	RO2.RXC	RO2.RXC	Peroxy radical reactions
HCHO.RXN	HCHO.RXN	HCHO.RXN	Reactions of formaldehyde
ALKPRODS.RXN	ALKPROES.RXC	ALKPRODS.RXC	Reactions of other alkane and alkene products
AROPRODS.RXN	AROPRODS.RX1	AROPRODS.RX2	Reactions of aromatic products
ETHENE.RXN	ETHENE.RXC	ETHENE.RXC	Reactions of ethene
ALKARO1.RXN ALKARO2.RXN (etc)	ALKARO1.RX1 ALKARO2.RX1 (etc)	ALKARO1.RX2 ALKARO2.RXN (etc)	Reactions of lumped alkanes and/or aromatics. One such file for each alkane and/or aromatic group.
LMPOLE1.RXN (etc)	LMPOLE1.RXC (etc)	LMPOLE1.RXC (etc)	Reactions of lumped alkenes. One such file for each lumped alkene group.

(continued)

Table 20 (continued) - 2

File Names			Description
Detailed Mechanism	Lumping Level #1	Lumping Level #2	
			Mechanism-Dependent PREPEMIT Program Input Files
LMPAE221.LPC	LMPCE221.LPC	LMPDE221.LPC	Lumping control files for mechanisms with two lumped alkanes, two lumped aromatics, ethene and one higher alkene.
LMPAE433.LPC	(a)	(a)	Lumping control files for mechanisms with four lumped alkanes, three lumped aromatics, ethene and three lumped higher alkenes.
ALKANE.LPM	ALKANE.LPM	ALKANE.LPM	Alkane mechanistic parameters
OTHERS.LPM	OTHERS.LPM	OTHERS.LPM	Mechanistic parameters for miscellaneous compounds which are generally lumped with alkanes.
AROMATIC.LPM	AROMATIC.LPM	AROMATIC.LP2	Aromatic Mechanistic Parameters
AALKENE.LPM	AALKENE.LPM	AALKENE.LPM	Mechanistic parameters for non-biogenic alkenes.
BIOALKE.LPM	BIOALKE.LPM	BIOALKE.LPM	Mechanistic parameters for biogenic alkenes.
EXPLICIT.LPM	EXPLICIT.LP1	EXPLICIT.LP2	Specifications of the reactive intermediate species which are explicitly represented.
LUMPMOLE.LPM	LUMPMOLE.CLP	LUMPMOLE.CLP	Name substitutions and "lumped molecule" representations of detailed species.
LUMPMECH.LPM	LUMPMECH.LPM	LUMPMECH.LPM	Translation table for mechanistic model species which are lumped with the model species which are assumed to have the same mechanism.

[a] File not on the distribution, but can be readily prepared by suitable modification of similar files which are. See comments in the files using the same lumping level and/or numbers of lumped groups.

emitted chemicals or their reaction products. On the other hand, for other applications, such as calculations assessing differences in reactivities of individual organics, or predictions of formation of secondary organic pollutants (most of which are lumped with other pollutants in the highly condensed mechanisms) larger, more chemically accurate mechanisms, may be required.

The users of this series of mechanisms have several ways in which they can control the degree of chemical detail which can be represented in the model simulation. These are as follows:

(1) As discussed in Section III, the numbers of "lumped groups" which are utilized to represent the reactions of the emitted alkanes, alkenes, or aromatics can be varied. As few as one lumped group or as many as nine can be employed to represent the reactions of each type of these species. The test calculations discussed in this section illustrate use of several options in this regard.

(2) The user has three options, or mechanisms, which differ in the number of species used to represent the reactions of the reactive organic products. The first option is to utilize the mechanism with the "detailed" representation, where 18 species are utilized to represent these products. The other two options are referred to as the "condensation level 1" and "condensation level 2" mechanisms, which utilize 10 and 8 species, respectively, to represent these products. The species used to represent the reactive organic products in these three mechanisms are summarized in Table 21. Input files implementing all three of these options are included in the distribution, and the test calculations are based on two of these options. The differences between the model predictions utilizing there various mechanisms are described in more detail elsewhere (Carter, 1988).

(3) The user has the option whether to include the reactions of H_2O_2 and "-OOH," the lumped species used to represent reactions at organic hydroperoxy groups, in the mechanism. The reactions of these species must be included in applications which are concerned with accurate predictions of hydroperoxy species (such as models for predictions of acid deposition), but these have relatively little effect on predictions of ozone. Therefore, for applications concerned primarily with ozone predictions, these reactions can be deleted, which effectively remove two species and

five reactions from the mechanism. Whether these reactions are included are controlled by the presence or absence of the record ". OOHRXN" in the preparation program input file prior to the inclusion of the mechanism input files, as indicated below. The test calculations include those both with and without the H_2O_2 or -OOH reactions.

Table 21. Species Used to Represent Reactive Organic Products in the Chemical Mechanisms

Model Species Name: Detailed Mechanism	Model Species Name: Lumping Level #1	Model Species Name: Lumping Level #2	Description
HCHO	HCHO	HCHO	Formaldehyde
CCHO	CCHO	CCHO	Acetaldehyde or higher aldehydes
RCHO			Higher aldehydes
ACET			Acetone
MEK	MEK	MEK	Higher Ketones or all ketones
RNO3	RNO3	RNO3	Alkyl nitrates
PAN	PAN	PAN	PAN or all PAN analogues
PPN			Higher PAN analogues
GPAN			PAN analogue formed from glyoxal
PBZN			Peroxy benzoyl nitrate
BALD			Aromatic Aldehydes
PHEN			Phenol
CRES	CRES	CRES	Alkyl phenols or all phenols
NPHE			Nitrophenols
GLY	GLY		Glyoxal
MGLY	MGLY	MGLY	Methyl glyoxal
AFG1	AFG1		Uncharacterized aromatic fragmentation product #1
AFG2	AFG2	AFG2	Uncharacterized aromatic fragmentation product #2

(4) The user can control whether or not the reactions of SO_2 are in the mechanism. This is determined by whether or not the preparation program includes the record "@SO2.RXN," which causes input to be taken from the file SO2.RXN which gives the reactions of SO_2, and causes it to be added to the list of active species. The reactions of SO_2 are included in all the preparation program input files in this distribution, but they can be readily removed by removing this record in the preparation input files.

The test calculations whose output files are included in this distribution employ three sets of options concerning the sets of reactive species and number of lumped groups employed. These are as follows:

The LMPAE433 Mechanism. The examples employing the most detailed representation of the emitted organics and the reactive intermediates employ the version of the mechanism designated "LMPAE433." (The prefix "LMP" is used to indicate that this uses the emissions lumping approach described in Section III; the "A" refers to the set of reactive intermediates employed, as indicated in Table 20; the "E" indicates that ethene is represented explicitly, and the "433" indicates the numbers of lumped groups used for alkanes, aromatics, and higher alkenes, respectively.) The main preparation program input file for this mechanism (LMPAE433.PRP) is listed in Table 22, and the lumping control file which should be used when processing emissions for this mechanism (LMPAE433.LPC) is listed in Table 23. The listing of these files indicate which mechanism module (.RXN) or mechanistic parameter (.LPM) input files that are used in the preparation of this mechanism or the processing of emissions data for it. This mechanism has 44 active species and 165 reactions. This mechanism probably has a greater level of detail than required in most applications, but is useful as a standard against which the predictions of more condensed mechanisms can be assessed.

The LMPAE221 Mechanism This mechanism employs the detailed representation of the reactive intermediates, but a more condensed representation of the emitted organics. It is thus the same as the "LMPAE433" mechanism, except that it uses two lumped groups for the alkanes, two for the aromatics, and one for the higher alkenes, rather than four, three, and three, respectively. The preparation program input file for this mechanism (LMPAE221.PRP) is similar to that shown for "LMPAE433" in Table 22,

Table 22. Listing of LMPAE433.PRP, the Main Preparation Program Input File for the LMPAE433 Mechanism

```
MODEL = LMPAE433.  DETAILED PRODUCTS, 7 ALKAROS, ETHENE, 3 LMPOLE.
!
!       Parameter input:  Default Temperature is 300 K.
!
TEMP=300.
!
!       End parameter input.
.
!
!       Rate constants need to be converted from cm, molec, sec units
!       to ppm, min units.
!
.UNITS=PPM
!
!       The reactions of H2O2 and -OOH are to be included.  (Delete
!       or "comment out" the following input if they are to be omitted.)
!
. OOHRXN
!
!       Mechanism input files.
!
@NOXAIR.RXN
@RO2.RXN
!
@HCHO.RXN
@ALKPRODS.RXN
@AROPRODS.RXN
!
@ALKARO1.RXN
@ALKARO2.RXN
@ALKARO3.RXN
@ALKARO4.RXN
@ALKARO5.RXN
@ALKARO6.RXN
@ALKARO7.RXN
!
@ETHENE.RXN
@LMPOLE1.RXN
@LMPOLE2.RXN
@LMPOLE3.RXN
!
!       If reactions of SO2 are not needed, delete or "comment out" the
!       following input.
!
@SO2.RXN
!
.END
```

Table 23. Listing of LMPAE433.LPC, the Lumping Control File Used for Processing of Emissions Input when Using the LMPAE433 Mechanism

---

```
!       File = LMPAE433.LPC
!
!       Lumping control file for model version 4/88a
!
!       Four lumped alkanes, three lumped aromatics, ethene and
!       three lumped higher alkene.
!

!
!       Mechanistic parameters are to be calculated for 300 K.
!
TEMP 300.0
!
!       For lumped groups where rate constant weighing of emissions is
!       employed, the fractions reacted is to be calculated assuming
!       an integrated OH radical level of 50 ppt-min.
!
INTOH   50.
!
!       Specifications controlling lumping of alkanes:  Four classes
!       are used, based on their rates of reaction with OH radicals.  For
!       the least reactive class, those reacting slower than 5.0E+3 ppm-1
!       min-1, rate constant weighing is used.  The alkanes in the other
!       three classes are lumped without weighing.
!
!       The first four of the seven alkane/aromatic lumped groups
!       in this mechanism are used for the alkanes.
!
GROUP 1 AAR1 A1 1
GROUP 1 AAR2 A2 0    5.0E+3
GROUP 1 AAR3 A3 0    1.0E+4
GROUP 1 AAR4 A4 0    1.7E+4
!
!       The mechanistic parameters for species lumped as alkanes
!       are taken from the following lumping control files.  The
!       "ALKANE" file contains the alkane parameters produced by
!       the ALKANE program.  The "OTHERS" file contains the parameters
!       assigned for non-alkane species which are generally lumped
!       with alkanes, such as alcohols, ethers, etc.  Note that these
!       two mechanistic parameter files have different data formats.
!
PARM 1 2 ALKANE.LPM
PARM 1 1 OTHERS.LPM
!
!       Specifications of lumping of the Aromatics:  Those reacting slower
!       than 2.0E+4 (benzene and mono-alkylbenzenes) are in the first
!       lumped aromatic class, those reacting with a range of rate constants
!       appropriate for the dialkylbenzenes (e.g., xylenes) are in the
```

(continued)

```
!      second lumped class, and those reacting faster than 5.0E+4 ppm-1
!      min-1, generally naphthalenes and tri- and polyalkylbenzenes,
!      are lumped in the most reactive class.  Rate constant weighing is
!      used for the least reactive class, with the OH rate constant forced
!      to be that appropriate for toluene.  No rate constant weighing is
!      used for the more reactive classes.
!
!      The last three of the seven alkane/aromatic lumped groups
!      in this mechanism are used for the aromatics.
!
GROUP 2 AAR5 A5 1    0.0       9.0E+3
GROUP 2 AAR6 A6 0    2.0E+4
GROUP 2 AAR7 A7 0    5.0E+4
!
!      The mechanistic parameters for the aromatics are given
!      in the following file.  Note that the mechanistic parameter
!      file for the aromatics depends on the level of detail used
!      to represent the reactive aromatic products.  The AROMATIC.LPM
!      file is appropriate for the most detailed representation.
!
PARM 2 1 AROMATIC.LPM
!
!      Ethene is represented explicitly.  The "REPLACE" input
!      is used to convert the detailed model species name "ETHENE"
!      to the name used for it in the lumped mechanism.  The "SPECIES"
!      input indicates that this is a lumped model species, and assigns
!      a number of carbons and molecular weight to it.
!
REPLACE
ETHENE  ETHE
END
!
SPECIES
ETHE     2   28.05
END
!
!      Three lumped groups are used for all the higher alkenes.  The
!      first uses a rate constant range appropriate for the 1-alkenes
!      such as propene, 1-butene, etc.  The second uses a rate constant
!      range appropriate for the iso-alkenes, and the third is used
!      for the internal alkenes.  No rate constant weighing is used, since
!      the non-ethene alkenes react relatively rapidly.
!
GROUP 3 OLE1 O1 0
GROUP 3 OLE2 O2 0    6.0E+4
GROUP 3 OLE3 O3 0    8.0E+4
!
!      The files AALKENE.LPM and BIOALKE.LPM contain the mechanistic
!      parameters for the anthropogenically and biogenically emitted
```

(continued)

```
!       alkenes, respectively.  These are given in separate files to
!       permit them to be represented by separate lumped groups, as
!       might be appropriate in model applications with significantly
!       different spacial distributions of emissions of such species.
!       However, they are lumped together for this model.
!
PARM 3 3 AALKENE.LPM
PARM 3 3 BIOALKE.LPM
!
!       Input for "Lumped molecule" and name substitutions are taken
!       from the following two files.
!
!       LUMPMOLE.LPM gives the name substitutions implementing the lumped
!       molecule representations used for the mechanism with the detailed
!       representation of the reactive products.
!
REPLACE LUMPMOLE.LPM
!
!       LUMPMECH.LPM translates the names used for generic mechanism
!       species to those for the species whose mechanisms are used to
!       represent them.
!
REPLACE LUMPMECH.LPM
!
!       The file EXPLICIT.LPM gives the list of explicit model species
!       used to represent the reactive products in the more detailed
!       version of the mechanism.
!
SPECIES EXPLICIT.LPM
!
!       End of file.
```

except that the references to the files ALKARO5, ALKARO6, ALKARO7, LMPOLE3, and LMPOLE3 are eliminated. The listing of the lumping control file for use when processing emissions data for this mechanism (LMPAE221.LPC) is given in Table 24. This mechanism has 39 active species and 154 reactions. This is a level of detail that might be considered to be most appropriate to reactivity assessment calculations.

Table 24. Listing of LMPAE221.LPC, the Lumping Control File Used for Processing of Emissions Input when Using the LMPAE221 Mechanism

---

```
!       File = LMPAE221.LPC
!
!       Lumping control file for model version 4/88a
!
!       Two lumped alkanes, two lumped aromatics, ethene and
!       one lumped alkene.
!
!
!       Mechanistic parameters are to be calculated for 300 K.
!
TEMP 300.0
!
!       For lumped groups where rate constant weighing of emissions is
!       employed, the fractions reacted is to be calculated assuming
!       an integrated OH radical level of 50 ppt-min.
!
INTOH   50.
!
!       Specifications controlling lumping of alkanes:  Those reacting
!       slower than 1.0E+4 ppm-1 min-1 are lumped in the slower class,
!       with rate constant weighing, and those reacting faster are lumped
!       in the second class, without rate constant weighing.  For SCAB
!       emissions, this amounts to approximately equal OH reactivity in
!       each lumped alkane group.
!
!       The first two of the four alkane/aromatic lumped groups
!       in this mechanism are used for the alkanes.
!
GROUP 1 AAR1 A1 1
GROUP 1 AAR2 A2 0    1.0E+4
!
!       The mechanistic parameters for species lumped as alkanes
!       are taken from the following lumping control files.  The
!       "ALKANE" file contains the alkane parameters produced by
!       the ALKANE program.  The "OTHERS" file contains the parameters
!       assigned for non-alkane species which are generally lumped
!       with alkanes, such as alcohols, ethers, etc.  Note that these
!       two mechanistic parameter files have different data formats.
!
PARM 1 2 ALKANE.LPM
PARM 1 1 OTHERS.LPM
!
!       Specifications of lumping of the aromatics:  Those reacting slower
!       than 2.0E+4 (benzene and mono-alkylbenzenes) are in the first
!       lumped aromatic class, the more reactive ones in the second.  Rate
!       constant weighing is used for the less reactive class, with the
!       OH rate constant forced to be that appropriate for toluene.  No
!       rate constant weighing is used for the more reactive class.
```

(continued)

```
!
!      The second two of the four alkane/aromatic lumped groups
!      in this mechanism are used for the aromatics.
!
GROUP 2 AAR3 A3 1    0.0       9.0E+3
GROUP 2 AAR4 A4 0    2.0E+4
!
!      The mechanistic parameters for the aromatics are given
!      in the following file.  Note that the mechanistic parameter
!      file for the aromatics depends on the level of detail used
!      to represent the reactive aromatic products.  The AROMATIC.LPM
!      file is appropriate for the most detailed representation.
!
PARM 2 1 AROMATIC.LPM
!
!      Ethene is represented explicitly.  The "REPLACE" input
!      is used to convert the detailed model species name "ETHENE"
!      to the name used for it in the lumped mechanism.  The "SPECIES"
!      input indicates that this is a lumped model species, and assigns
!      a number of carbons and molecular weight to it.
!
REPLACE
ETHENE   ETHE
END
!
SPECIES
ETHE     2   28.05
END
!
!      Only one lumped group is used for all the higher alkenes.
!      No rate constant weighing is used, since the non-ethene alkenes
!      react relatively rapidly.
!
GROUP 3 OLE1 O1 0
!
!      The files AALKENE.LPM and BIOALKE.LPM contain the mechanistic
!      parameters for the anthropogenically and biogenically emitted
!      alkenes, respectively.  These are given in separate files to
!      permit them to be represented by separate lumped groups, as
!      might be appropriate in model applications with significantly
!      different spacial distributions of emissions of such species.
!      However, they are lumped together for this model.
!
PARM 3 3 AALKENE.LPM
PARM 3 3 BIOALKE.LPM
!
!      Input for "Lumped molecule" and name substitutions are taken
!      from the following two files.
```

(continued)

Table 24 (continued) - 3

```
!
!      LUMPMOLE.LPM gives the name substitutions implementing the lumped
!      molecule representations used for the mechanism with the detailed
!      representation of the reactive products.
!
REPLACE LUMPMOLE.LPM
!
!      LUMPMECH.LPM translates the names used for generic mechanism
!      species to those for the species whose mechanisms are used to
!      represent them.
!
REPLACE LUMPMECH.LPM
!
!      The file EXPLICIT.LPM gives the list of explicit model species
!      used to represent the reactive products in the more detailed
!      version of the mechanism.
!
SPECIES EXPLICIT.LPM
!
!      End of file.
```

The LMPBE221 Mechanism This mechanism employs the least detailed representation of the reactive intermediates, but uses the same number of species to represent the emitted organics as the LMPAE221 mechanism. The LMPBE221 mechanism employs the "level 2" condensation of the reactive organic intermediates (as indicated in Table 20), and also neglects the reactions of H_2O_2 and lumped organic hydroperoxides. The listing of the preparation program input file for this mechanism (LMPBE221.PRP) is given in Table 25. The lumping control file for this mechanism (LMPBE221.LPC) is the same as that used for the LMPAE221 mechanism, except that it uses the file "AROMATIC.LP2" instead of "AROMATIC.LPM" to obtain the mechanistic parameters for the aromatics. This mechanism has 27 active species and 102 reactions. This is a level of detail recommended for airshed model applications where ozone or sulfate predictions are of major interest. Because it neglects hydroperoxide reactions, this mechanism is not appropriate for acid deposition models. However, the comments in the LMPBE221.PRP listing in Table 25 indicates how it can be changed so that these reactions would be represented. This would add two species and five reactions to the overall mechanism.

Table 25. Listing of LMPBE221.PRP, the Main Preparation Program Input File for the LMPBE221 Mechanism

```
MODEL=LMPBE221.  LUMP LEVEL 2;  4 ALKAROS, 2 ALKENES.  NO XOOH
!
!       Parameter input:  Default Temperature is 300 K.
!
TEMP=300.
!
!       End parameter input.
.
!
!       Rate constants need to be converted from cm, molec, sec units
!       to ppm, min units. !
.UNITS=PPM
!
!       The reactions of H202 and -OOH are NOT included.  To include
!       them, delete the "!" character preceding the ". OOHRXN" input
!       line.  This will cause conditional input for H2O2 and -OOH
!       reactions to be processed.
!
!. OOHRXN
!
!       Mechanism input files.
!
@NOXAIR.RXN
@RO2.RXC
!
@HCHO.RXN
@ALKPRODS.RXC
@AROPRODS.RX2
!
@ALKARO1.RX2
@ALKARO2.RX2
@ALKARO3.RX2
@ALKARO4.RX2
!
@ETHENE.RXN
!
@LMPOLE1.RXC
!
!       If reactions of SO2 are not needed, delete or "comment out" the
!       following input.
!
@SO2.RXN
!
.END
```

B. Emissions Assignments Implemented in the Data Files

As discussed in Section III, this software is based on the assumption that the emissions input data will be given in terms of mass emissions of SAROAD classes, which the emissions processing programs then convert to molar emissions of detailed model species, which in turn are then converted into molar emissions of the lumped species in the mechanism. By "emissions assignments" we thus refer to the assignments of detailed model species to the various SAROAD classes. These assignments are given in the file SAROAD.PRM, which is included with this distribution. This file contains assignments of detailed model species to all SAROAD classes which were found to have non-zero emissions in the 1983 total California emissions profiles we received from the California ARB. (There are SAROAD classes on the data base used by the CARB where there were zero emissions in the 1983 inventory, and assignments have not been made for most of these.)

The emissions assignments developed for this program were also used to process emissions data from the EPA/NAPAP inventory for the entire contiguous United States. This inventory has a much larger number of SAROAD classes with non-zero emissions, and assignments for many of these have not been made. However, the classes for which assignments which have been made account for over 96% (by weight) of the U.S. emissions, so this is not regarded as a serious problem. (To assign the other 4% would require making assignments for hundreds of additional SAROAD classes, no one of which makes a significant contribution by itself.) A more serious problem is that there are approximately 30 SAROAD classes, some with non-negligible emissions, which have different chemicals associated with them in the EPA/NAPAP profiles than those used by the CARB (Carter, 1988). For this reason, we also include in this distribution a data set called SAROAD.EPA, which consists of the SAROAD assignments appropriate for use with emissions data from the EPA/NAPAP data base. When processing such data, the present SAROAD.PRM data set should be removed or renamed, and the SAROAD.EPA data set renamed to SAROAD.PRM. However, all of the example calculations discussed in this section utilize CARB emissions data, and thus the SAROAD.EPA data set was not used.

As discussed elsewhere (Carter, 1988), many of the assignments in the SAROAD.PRM and SAROAD.EPA data sets are uncertain. This is not only because there are emitted chemicals whose atmospheric reaction mechanisms are unknown, and thus may not be well represented by any of the detailed model species for which we have mechanistic estimates, but also because there are SAROAD classes whose chemical compositions are uncertain or ambiguous. In addition, because several hundred assignments had to be made in a relatively limited amount of time, the possibility that there may be minor errors in the present version of the SAROAD.PRM file cannot be entirely ruled out. It is recommended that the contents of this file be independently reviewed and checked for errors (such as in molecular weights, etc.) before it is routinely used for airshed model applications. While serious errors are not anticipated, this type of quality control is appropriate, but could not be carried out within the limited scope of this program.

C. Test Calculations

The data sets included in this distribution include input and selected output files for a number of test calculations using the chemical mechanisms and emissions assignments discussed above. These calculations utilized the LMPAE433, LMPAE221, and the LMPBE221 mechanisms discussed above, and total California (CALTOT) and total South Coast Air Basin (SOUCOA) emissions profiles from the 1983 CARB inventory. For each mechanism, the Preparation program was run, then the emissions data were processed using the lumping control file appropriate for that mechanism and the CALTOT emissions as the representative profile, and then single cell model simulations were carried out using these data as input. The conditions of the model simulations, the specific procedure and input data used in carrying them out, and selected results obtained, are summarized in this section.

The model simulations employed a modified version of the SAPRC model calculation program, which was linked with the four airshed model interface subroutines discussed in Section IV.C, and with the mechanism-specific subroutines produced by the preparation program when processing the mechanism. A discussion of this program is beyond the scope of this document, and its major operations are only briefly summarized here. When first invoked and given the name of the input file to use, the program calls RDMOD to read the mechanism specific (.MOD) data files produced by the preparation program, then calls NEWRK to calculate the thermal rate constants for a temperature of 300 K, and then calls NEWPHK to calculate the photolysis rate constants for a solar zenith angle of zero. After reading the initial reactant concentrations from the input files, it then carries out the model simulation for the specified amount of time using the photolysis and thermal rate constants held constant at their initially calculated values (which are given in the output listing produced by RDMOD). A "Hybrid" solver similar to that used in the Caltech airshed model (Croes, 1987) was employed in these test simulations. The calculated concentration-time data were included in the output file, and are also saved in separate calculation data (.CDI) output files.

A representative model simulation input file used in these example calculations is listed in Table 26. Although the format of the input data used by this program is not documented here, the comments included in the file (indicated by a "!" character in the first position) should make the meaning of the input data employed reasonably obvious. (As with the preparation program described in Section II, a "@" in the first position of the input record means that input is taken from another file.) All these simulations were 12-hour static calculations with all the reactants present initially, with no dilution and with constant light intensity. The initial NO and NO_2 were 62.5 and 20.8 ppb, respectively, and the emissions input data were normalized to yield 1 ppmC total of non-methane and non-CO organics, resulting in an initial NMHC/NO_x ratio of 12. As indicated in Table 26, each input file resulted in two model simulations, the first using the processed CALTOT emissions profiles, and the second using the processed SOUCOA profiles. These emissions profiles were those which were directly output by the LUMPEMIT program, being produced as described below.

For each of these three mechanisms, the overall procedure in carrying out the test calculations were the same. This will be illustrated below in a step-by step manner for using the LMPBE221 model as the example. The same procedure was employed for the other two models.

(In the examples of computer interaction in the following discussion, the user input is underlined, and a "$" symbol is used to indicate system prompts. The notation "<CR>" will be used to indicate that the user just

Table 26. Listing of LMPBE221.INT, the Model Simulation Input File Used in the Test Calculations Employing the LMPBE221 Mechanism

```
TEST ARB MODEL.  STATIC SIMULATION WITH CONSTANT LIGHT INTENSITY
!
!       Indicate the mechanism used.  This tells the RDMOD
!       routine which .MOD file to use to obtain the mechanism-
!       dependent parameters.
!
MODEL=LMPBE221
!
!       Indicate the file which has the emissions-dependent parameters,
!       which was prepared using the CALTOT emissions profile and
!       the lumping control file appropriate for this mechanism.
!       This input tells the RDPRM routine which .RXP file to use
!       to obtain the emissions-dependent parameters.
!
RXP=CALBE221
!
!       The concentration-time data from this calculation will be saved
!       on disk.
!
SAVE
!
!       End parameter input.
!
.
!
!       Label the type of mechanism used.
!
.LBL1 Model=LMPBE221, E.Parms=CALTOT,
!
!       Data printed every hour.  Twelve-hour simulation
!
.DPRN  60.
.TO    0.0
.TEND  720.
!
!       Temperature is constant at 300 K in this scenario.  50% RH.
!
.TEMP  300.
H2O    2.08E+4
!
!       NOx level is 0.08 ppm, making the HC/NOx ratio 12 for
!       1 ppmC NMCH in the ROG mixture.
!
NO     0.0625
NO2    0.0208
!
!       This simulation has 0.1 ppm SO2
!
```

(continued)

Table 26 (continued) - 2

```
SO2     0.1
!
!       Above are defaults for the following two simulations, where
!       the input emissions profile is varied.
!
.SETDEF
!
!       Do the simulation for the CALTOT emissions profile:
!
!       Label indicating the emissions profile used
!
.LBL4 ROG mix=CALTOT
!
!       Input is taken from the .EMO file prepared for this profile
!       using the lumping processing for this mechanism.  This file
!       contains the initial concentrations of each of the lumped
!       species in the mechanism which represent the emitted organics.
!
@CALBE221.EMO
!
!       This command causes the integration to be carried out using
!       the data input thus far.
!
.INT
!
!       Do the simulation for the SOUCOA emissions profile.
!
.LBL4 ROG mix=SOUCOA
@SOUBE221.EMO
.INT
```

hits the "return" key at the prompt. The system command syntax used will be appropriate for DEC VAX/VMS systems.)

Step 1: The first step was to run the preparation program for the mechanism, and then link the mechanism-specific subroutines produced with the rest of the airshed modeling program. If the preparation program is given the name "PREP," the commands and prompts involved in running this program might be as follows:

```
$RUN PREP
NAME FOR INPUT/OUTPUT FILES = LMPBE221
MODEL = 'LMPBE221         ' PREPARED.
```

The following commands then might be used to compile the mechanism dependent subroutine produced and link it with the rest of the model simulation program.

```
$RENAME LMPBE221.FTN LMPBE221.FOR
$FOR LMPBE221
$LINK INTMAIN,LMPBE221,RDMOD,NEWRK,NEWPHK,RDPRM,TESTINT.OLB/LIB
$RENAME INTMAIN.EXE LMPBE221.EXE
```

(This example assumes that the "FOR" command will compile FORTRAN programs, that .FOR, rather than .FTN is the expected extension for FORTRAN source files, that the "LINK" command will link them, that INTMAIN is the main module for the simulation program, and all the modules of this program other than the mechanism-dependent subroutine and the interface routines documented here are in the object library TESTINT.OLB.) The resulting program is then renamed to LMPBE221 to indicate that it is to be used for this mechanism only.

Step 2: The next step is to run PREPEMIT to calculate the emissions dependent parameters, and to prepare for processing the emissions data for this mechanism. In the example calculations demonstrated here, the CALTOT emissions profile is used as the representative profile for deriving the emissions-specific parameters. (However, for a simulation of a specific airshed, the total emissions for the specific airshed would obviously be more appropriate for this purpose.) The CALTOT emissions input data are in two files, CALTOT.EMI and CALTOT.EMN, where the former give the total emissions amounts, while the latter have the emissions amounts normalized to yield a total of 1 ppmC of NMHC after being processed by LUMPEMIT. Either of these files could be used in the PREPEMIT run, since the emissions-dependent parameters depend only on the relative compositions of the profiles. For these examples, the unnormalized data, in the CALTOT.EMI file, are used in the PREPEMIT run.

The commands and prompts involved in running PREPEMIT for the LMPBE221 mechanism and the CALTOT profile for the purpose of these examples are as follows:

```
$RUN PREPEMIT
File name for control parms (<CR> to end): LMPBE221.LPC
Parameters for  44 substitutions and 145 model species read.
Assignments for 352 SAROAD species read.

Emissions file name: CALTOT.EMI
Output file for lumped parameters: CALBE221.RXP
Output file for lumped species, if different:  <CR>
Normalize to one mole NMHC (y/n, default=no)? Y
Output file for lumped SAROAD assignments: CALBE221.SAR

Emissions file name: <CR>
Done.
```

This causes the emissions-specific parameters to be output into the file CALBE221.RXP and the lumped SAROAD assignments to be output into the file CALBE221.SAR. The responses to the prompts for the lumped species output and asking whether they are to be normalized affect only the calculated amounts of the lumped species, which are not used in this example, since these are calculated separately using the LUMPEMIT program. (The output lumped species amounts are normalized so they can be compared with the output of the LUMPEMIT program for the CALTOT profile. They should be the same.)

This and the previous step can be thought of as being analogous to preparing a mechanism for use in simulations based on an assumed representative profile. If the mechanism or the profile taken as representative do not change, then these steps do not need to be repeated, and the output files produced can be utilized for any number of airshed model simulations. For the LMPBE221 mechanism and the CALTOT representative profile, the minimum number of mechanism- and emissions-dependent files required to process the detailed emissions input data and carry out the simulations are LMPBE221.EXE (the airshed modeling program with the specific mechanism built into it), LMPBE221.MOD (the mechanism specific data needed by the airshed modeling program), CALBE221.RXP (the emissions specific data needed by the airshed program), and CALBE221.SAR (the lumped SAROAD assignments needed by PREPEMIT to process the emissions data). The other files produced in the above steps, e.g., the .FTN, and .PRO files produced by the preparation program, and the .OBJ and .LIS files produced when the mechanism-specific subroutines are compiled, can be deleted.

Step 3: The next step is to use LUMPEMIT to process the emissions input data for the model simulations. For multi-cell airshed model applications, this would involve processing the emissions profiles for all the cells in the model, and for all the times in the model where the emissions profiles might be different. For these example calculations, only two profiles are used in each calculation, those being CALTOT and SOUCOA. They can be thought of as being analogous to emissions into specific cells of a model simulation. For these examples, the normalized profiles in CALTOT.EMN and SOUCOA.EMN are used, so the PREPEMIT output will yield total NMHC levels of 1 ppmC. (LUMPEMIT does not normalize the data it processes, since its purpose is to process absolute emissions amounts. In this example, the normalization was done by a separate program, and the normalized data are provided as input.) The commands and prompts involved in running LUMPEMIT for this example are as follows:

```
$RUN LUMPEMIT
Conversion factor (<CR> = use 1.0): <CR>
SAROAD assignment file: CALBE221.SAR
Output emissions amounts with species names in
  separate files (No)? Y
Emission set # 1: CALTOT.EMN
Emissions output set # 1: CALBE221.EMO
Emission set # 2: SOUCOA.EMN
Emissions output set # 2: SOUBE221.EMO
Emission set # 3: <CR>
```

This results in the files CALBE221.EMO and SOUBE221.EMO, containing the processed normalized emissions data for CALTOT and SOUCOA, respectively, being produced by PREPEMIT. As shown in Table 26, these files are directly read by the model simulation program to obtain the initial concentrations of the lumped species used in the mechanism to represent these emissions.

Note that in this example, the option to have the processed emissions output in separate files was chosen. This is because the SAPRC single-cell model simulation program requires the data to be in this format. However, for multi-cell airshed model simulations, where a large number of emissions profiles are to be processed, it would clearly be more efficient to use the default option of having the emissions data combined in a single file with the more condensed format, discussed in Section IV.B.3.c.

Step 4: The final step is to carry out the model simulation itself. For this example, this would involve the following operations and prompts:

```
$RUN LMPBE221
NAMES FOR INPUT/OUTPUT FILES = LMPBE221
INT LMPBE221  :   2 CALCS.  INT, TOT TIMES = (etc)
```

The latter message indicates that the program has completed, and that it carried out two simulations. The file LMPBE221.OUT is created. This output file contains a listing of the mechanism and the rate constants used (output by the RDMOD module during mechanism initialization), and the initial and calculated concentration-time data for all the species in the mechanism, output each simulated hour, for each of the two simulations. The program will also output "calculation data" files, giving the concentration-time data in a more condensed and more readily computer-readable format (for use when making plots of the results, etc.). In the example output files in the distributed data sets, these will be converted to .CDI files, one for each calculation. For the above example, the output of the first simulation, using the possessed CALTOT emissions data as input, the concentration time data will be in the file CALBE221.CDI, and the output of the second simulation, using the processed SOUCOA data, the output will be in the file CALBE221.CDI. The output files for the calculations using the other mechanisms are named analogously.

Selected results of these test calculations are given in Table 27. It can be seen that there is relatively little difference between the calculations using the different emissions profiles; this is not unexpected since the emissions profiles are very similar in their composition. The three models employed give reasonably similar predictions for a given mixture, though there are differences. These differences illustrate

Table 27. Selected Concentration-Time Data Obtained in the Example Calculations Using the LMPAE433, LMPAE221, and LMPBE221 Mechanisms and the CALTOT and SOUCOA Emissions Input Data

Time (min)	CALTOT			SOUCOA		
	LMPAE433	LMPAE221	LMPBE221	LMPAE433	LMPAE221	LMPBE221
			Ozone (ppm)			
120.0	5.8211E-02	6.1449E-02	5.7629E-02	6.1647E-02	6.5266E-02	6.1383E-02
240.0	0.1646	0.1730	0.1648	0.1681	0.1766	0.1685
360.0	0.2853	0.2907	0.2858	0.2862	0.2912	0.2868
480.0	0.3322	0.3337	0.3390	0.3298	0.3306	0.3371
600.0	0.3486	0.3537	0.3574	0.3458	0.3478	0.3576
720.0	0.3589	0.3668	0.3700	0.3558	0.3594	0.3713
			NO2 (ppm)			
120.0	5.2022E-02	5.2139E-02	5.2120E-02	5.2069E-02	5.2087E-02	5.2177E-02
240.0	3.8256E-02	3.5955E-02	3.7920E-02	3.7301E-02	3.4938E-02	3.6893E-02
360.0	1.1842E-02	1.0251E-02	1.2708E-02	1.1005E-02	9.5443E-03	1.1972E-02
480.0	3.4534E-03	3.8103E-03	3.8616E-03	3.3203E-03	3.3612E-03	3.8208E-03
600.0	2.8673E-03	3.2812E-03	3.3509E-03	2.7745E-03	2.9835E-03	3.4808E-03
720.0	2.6137E-03	2.9478E-03	3.1771E-03	2.5273E-03	2.7160E-03	3.2256E-03
			PAN (ppm)			
120.0	1.6500E-03	1.8027E-03	2.1524E-03	1.8109E-03	1.9973E-03	2.4120E-03
240.0	6.8442E-03	7.6586E-03	9.7075E-03	7.1397E-03	8.0030E-03	1.0225E-02
360.0	1.5355E-02	1.5897E-02	2.1344E-02	1.5653E-02	1.6190E-02	2.1915E-02
480.0	1.8050E-02	1.7990E-02	2.5222E-02	1.8000E-02	1.7789E-02	2.5354E-02
600.0	1.7210E-02	1.7824E-02	2.3167E-02	1.7102E-02	1.7178E-02	2.3850E-02
720.0	1.6297E-02	1.7352E-02	2.1514E-02	1.6164E-02	1.6574E-02	2.2286E-02
			HCHO (ppm)			
120.0	1.7971E-02	1.9367E-02	2.0034E-02	1.7730E-02	1.9135E-02	1.9979E-02
240.0	2.2586E-02	2.2759E-02	2.4924E-02	2.0936E-02	2.0988E-02	2.3353E-02
360.0	2.1515E-02	2.0625E-02	2.3584E-02	1.9494E-02	1.8584E-02	2.1662E-02
480.0	1.9646E-02	1.8926E-02	2.2480E-02	1.8330E-02	1.7709E-02	2.1302E-02
600.0	1.8644E-02	1.7693E-02	2.2011E-02	1.7892E-02	1.7154E-02	2.1098E-02
720.0	1.7328E-02	1.6431E-02	1.9982E-02	1.7022E-02	1.6274E-02	1.9683E-02
			H2SO4 (ppm)			
120.0	1.8306E-03	1.9016E-03	1.8052E-03	1.8888E-03	1.9723E-03	1.8748E-03
240.0	4.4776E-03	4.7309E-03	4.4878E-03	4.5166E-03	4.7859E-03	4.5416E-03
360.0	8.4353E-03	8.7646E-03	8.4337E-03	8.3828E-03	8.7219E-03	8.4006E-03
480.0	1.1610E-02	1.1898E-02	1.1808E-02	1.1387E-02	1.1670E-02	1.1623E-02
600.0	1.4264E-02	1.4725E-02	1.4659E-02	1.3931E-02	1.4295E-02	1.4443E-02
720.0	1.6940E-02	1.7583E-02	1.7642E-02	1.6490E-02	1.6960E-02	1.7365E-02

the effects of condensation of the mechanisms, and a comparison of the results of the calculations using the LMPAE221 mechanism to those using LMPAE433 show the effects of condensing the numbers of species used to represent emitted species, while a comparison of calculations using LMPAE221 with those using LMPBE221 show the effects of condensing the number of species used to represent reactive intermediates. For example, the LMPBE221 mechanism results in predictions of significantly more PAN formation than the mechanisms using the more detailed representation of the reactive products; this is because the mechanisms with the more condensed representation of reactive products used PAN to represent all acyl peroxynitrates, while in the more detailed mechanism, PAN represents only itself. The effects of various types of condensations are discussed in more detail elsewhere (Carter, 1988). However, these examples should illustrate how alternative condensation approaches could be employed, and how the appropriate level of condensation for the particular applications can be implemented.

VI. REFERENCES

Atkinson, R. (1988), 'Gas-Phase Atmospheric Chemistry of Organic Compounds", Appendix A of Final Report on California ARB Contract No. A5-122-32, October.

Carter, W. P. L., F. W. Lurmann, R. Atkinson, and A. C. Lloyd (1986), "Development and Testing of a Surrogate Species Chemical Reaction Mechanism," EPA-600/3-86-031, August.

Carter, W. P. L., A. M. Winer, R. Atkinson, S. E. Heffron, M. P. Poe, and M. A. Goodman (1987), "Atmospheric Photochemical Modeling of Turbine Engine Fuels. Phase II. Computer Model Development," Final Report on USAF Contract no. F08635-83-0278, Engineering and Services Laboratory, Air Force Engineering and Services Center, Tyndall Air Force Base, Florida, August.

Carter (1988). "Documentation of a Gas-Phase Photochemical Mechanism for Use in Airshed Modeling," Appendix B of Final Report on California ARB Contract A5-122-32, October.

Croes, B. (1987), California Air Resources Board Modeling Group, Sacramento CA, Private communication.

Lurmann, F. W., W. P. L. Carter, and Lori A. Coyner (1987), "A Surrogate Species Chemical Reaction Mechanism for Urban-Scale Air Quality Simulation Models. Volume I - Adaptation of the Mechanism", Final Report on EPA Contract No. 68-02-4104, Atmospheric Sciences Research Laboratory, Research Triangle Park, NC.

NASA (1987) "Chemical Kinetics and Photochemical Data for Use in Stratospheric Modeling. Evaluation Number 8," JPL Publication 87-41, Jet Propulsion Laboratory, Pasadena, CA, September.

Peterson, J. T. (1976) "Calculated Actinic Fluxes (290 - 700 nm) for Air Pollution Photochemistry Applications", EPA-600/4-76-025, June.

APPENDIX A

COMPUTER FILES IMPLEMENTING THE SOFTWARE AND EXAMPLE CALCULATIONS

The computer data files giving the FORTRAN source code for the programs described in this document, and the input and output files for the example calculations have been submitted to the CARB along with this document, and are available from the author upon request. The files are listed in Table A-1. These files are distributed on six 5 1/2" IBM-PC readable floppy diskettes, as indicated on Table A-1. The files are also available on 1600 BPI magnetic tape for the cost of duplication. If the files are distributed on tape, they will be arranged in the order indicated in Table A-1, and have fixed length records, with logical record lengths (LRECL's) of 80 or 132, as indicated on the table.

Table A-1. List of Source and Data Files Used in the Software Distribution and Example Calculations

No.	File Name	LRECL	Description
			Documentation (Disk #1, Main directory)
1	FILES.DOC	80	List of files in the distribution.
			Preparation Program Source Code (Disk #1, PREPPGM subdirectory)
2	PSPECS.FTN	80	Specifications for preparation program variables
3	NEWIOBUF.FTN	80	Specifications for I/O buffers and parameters
4	PREP.FTN	80	Preparation program Main routine
5	RDRXN.FTN	80	Subroutines for input processing and reaction
6	RXLST1.FTN	80	
7	RXLST2.FTN	80	
8	RXSUBS.FTN	80	
9	REORDR.FTN	80	Subroutines for processing and listing of species
10	LISTSC.FTN	80	
11	PNHRXN.FTN	80	Subroutines for mechanism dependent code preparation
12	BLDUP.FTN	80	
13	DIFFUN.FTN	80	
14	INSCOD.FTN	80	
15	VCOEF.FTN	80	
16	ASMD.FTN	80	
17	SAVERX.FTN	80	Subroutine for creation of the mechanism data file
18	PFALIN.FTN	80	Utility subroutines
19	NEWSUBS.FTN	80	
20	NEWINBUF.FTN	80	
			PREPEMIT Emissions Preparation Program Sources (Disk #1, PREPEMIT subdirectory)
21	LUMPSPEC.FTN	80	General specifications for PREPEMIT
22	SARSPEC.FTN	80	Specifications for detailed SAROAD assignments
23	PREPEMIT.FTN	80	Main program for PREPEMIT
24	LDSAR.FTN	80	Subroutines to read SAROAD.PRM
25	LMPEMI.FTN	80	Subroutine to input emissions data
26	RDLUMP.FTN	80	Subroutine to read lumping control files
27	RDGENP.FTN	80	Subroutines to read mechanistic parameter files
28	RDALKP.FTN	80	

Table A-1 (continued) - 2

No.	File Name	LRECL	Description
29	RDOLEP.FTN	80	
30	LMPSUB.FTN	80	Subroutine to derive lumped parameters
31	SARSUB.FTN	80	Subroutine to derive lumped SAROAD assignments
			LUMPEMIT Example Emissions Processing Sources (Disk #1, LUMPEMIT subdirectory)
32	LSARSPEC.FTN	80	Specifications for LUMPEMIT
33	LUMPEMIT.FTN	80	Main module for LUMPEMIT
34	SARINI.FTN	80	Subroutine to initialize SAROAD assignment data
35	LMPSAR.FTN	80	Subroutine to make lumped assignments
			Example Airshed Model Subroutines for Implementing Prepared Mechanisms (Disk #1, INTSUBS subdirectory)
36	MODLSPC.FTN	80	Common parameter and data specifications
37	RDMOD.FTN	80	Routine to input mechanism specific data
38	RDPRM.FTN	80	Routine to input emissions specific data
39	NEWRK.FTN	80	Routine to calculate thermal rate constants
40	NEWPHK.FTN	80	Routine to calculates photolysis rate constants
			Example Preparation Program Input Files (Disk #2)
41	LMPAE221.PRP	80	Main Preparation Program Input Files for mechanisms using two lumped alkanes, two lumped aromatics, and ethene + one lumped higher alkene. Different files for different condensation levels of the reactive LMPA = most detailed; LMPC = level 1 condensation; = level 2 condensation; LMPB = level 2 condensation with reactions of H2O2 and -OOH removed.
42	LMPCE221.PRP	80	
43	LMPDE221.PRP	80	
44	LMPBE221.PRP	80	
45	LMPAE433.PRP	80	Main Preparation Program input for a mechanism using four lumped alkanes, three lumped aromatics, ethene + 3 lumped higher alkenes and the most least condensed representation of the reactive products.

Table A-1 (continued) - 3

No.	File Name	LRECL	Description
			Mechanism Modules Used for Preparation Program Input for the most detailed representation of thr reactive products. (Mechanism version 4/88a.) Those indicated with a (*) are also used for the more condensed mechanisms.
46	NOXAIR.RXN	80	Inorganic reactions (*)
47	SO2.RXN	80	Reactions of SO2 (*)
48	RO2.RXN	80	General peroxy radical reactions
49	HCHO.RXN	80	Reactions of formaldehyde (*)
50	ALKPRODS.RXN	80	Reactions of other alkane and alkene products
51	AROPRODS.RXN	80	Reactions of aromatic products
52	ALKAROS.RXN	80	Reactions of the lumped groups used for alkanes and/or aromatics. (ALKAROS.RXN is the "template" file used to create the others. This is done by global substitution of the "?" character in the file with "1", "2", etc., using a text editor.)
53	ALKARO1.RXN	80	
54	ALKARO2.RXN	80	
55	ALKARO3.RXN	80	
56	ALKARO4.RXN	80	
57	ALKARO5.RXN	80	
58	ALKARO6.RXN	80	
59	ALKARO7.RXN	80	
60	ETHENE.RXN	80	Reactions of ethene (if treated explicitly)
61	LMPOLES.RXN	80	Reactions of lumped groups used for alkenes. (LMPOLES.RXN is the "template" file used to create the others.)
62	LMPOLE1.RXN	80	
63	LMPOLE2.RXN	80	
64	LMPOLE3.RXN	80	
			Mechanism modules used for preparation program input for mechanisms with more condensed representations of the reactive products. (Mechanism versions: Lumping level 1 = C4/88a; level 2 = C4/88b)
			Input files used by both lumping levels.
65	RO2.RXC	80	General peroxy radical reactions.
66	ALKPRODS.RXC	80	Reactions of alkane and alkene products other than formaldehyde.
67	LMPOLES.RXC	80	Reactions of lumped groups used for alkenes.
68	LMPOLE1.RXC	80	
			Input files used by lumping level 1.

Table A-1 (continued) - 4

No.	File Name	LRECL	Description
69	AROPRODS.RX1	80	Aromatic products
70	ALKAROS.RX1	80	Alkane/aromatic lumped groups
71	ALKARO1.RX1	80	
72	ALKARO2.RX1	80	
73	ALKARO3.RX1	80	
74	ALKARO4.RX1	80	
			Input files used by lumping level 2.
75	AROPRODS.RX2	80	Aromatic products
76	ALKAROS.RX2	80	Alkane/aromatic lumped groups
77	ALKARO1.RX2	80	
78	ALKARO2.RX2	80	
79	ALKARO3.RX2	80	
80	ALKARO4.RX2	80	
			Emissions Processing Input Files (Disk #3)
			Lumping Control Files
81	LMPAE221.LPC	80	Lumping control files based on using two lumped alkanes, two lumped aromatics, and ethene + one lumped higher alkene. Different files are used for the different condensation levels of the reactive intermediates, as indicated above in the listing of the corresponding preparation input files. (The LMPDE221 mechanism uses the LMPBE221 lumping control file.)
82	LMPCE221.LPC	80	
83	LMPBE221.LPC	80	
84	LMPAE433.LPC	80	Lumping control file based on using four lumped alkanes, two lumped aromatics, and ethene + three lumped alkenes, and the most detailed representation of the reactive intermediates.
			Emissions Assignments File
85	SAROAD.PRM	80	Assignments of detailed model species to SAROAD classes (for use with CARB emissions data)
86	SAROAD.EPA	80	Assignments of detailed model species to SAROAD classes (for use with EPA/NAPAP emissions data)
			Mechanistic Parameter Files
87	ALKANE.LPM	80	Mechanistic parameters for alkanes

Table A-1 (continued) - 5

No.	File Name	LRECL	Description
88	AROMATIC.LPM	80	Mechanistic parameters for aromatics, used for the detailed representation of the reaction products and for lumping level 1.
89	AROMATIC.LP2	80	Mechanistic parameters for aromatics, used for lumping level 2.
90	OTHERS.LPM	80	Mechanistic parameters for miscellaneous compounds which are generally lumped with alkanes
91	AALKENE.LPM	80	Mechanistic parameters for anthropogenically emitted alkenes
92	BIOALKE.LPM	80	Mechanistic parameters for biogenic alkenes
			Other Lumping Control Specification Files
93	EXPLICIT.LPM	80	Specifications of explicit lumped model species used for the detailed representation of reactive intermediates.
94	EXPLICIT.LP1	80	Specifications of explicit lumped model species used for lumping level 1.
95	EXPLICIT.LP2	80	Specifications of explicit lumped model species used for lumping level 2.
96	LUMPMOLE.LPM	80	Name substitutions and "lumped molecule" representations of detailed species, used for the detailed representation of reactive intermediates.
97	LUMPMOLE.CLP	80	Name substitutions and "lumped molecule" representations of detailed species, used for the lumped representations of the reactive intermediates (both lumping levels)
98	LUMPMECH.LPM	80	Translation table for mechanistic model species which are lumped with the model species which are assumed to have the same mechanism
			Example Emissions Input Files (Data supplied by the California Air Resources Board for from the Preliminary 1983 Inventory for Selected Air Basins. Two files are given for each set. The name.EMI files give emissions as lbs/day into the basin, and the name.EMN files give emissions in grams, normalized to yield 1 mole carbon, not counting CO or methane.
99	CALTOT.EMI	16	Total Emissions into California
100	SOUCOA.EMI	16	Total South Coast Air Basin Emissions
101	CALTOT.EMN	16	Normalized Emissions into California
102	SOUCOA.EMN	16	Normalized South Coast Emissions

Table A-1 (continued) - 6

No.	File Name	LRECL	Description
			Test Calculation Input and Output Files (Disk 4: LMPAE433 Model; Disk 5: LMPAE211 Model; Disk 6: LMPBE211 Model)
			Integration test input files
103	LMPAE433.INT	80	Integration test input files files which are associated with the mechanism input using the .PRP file of the same name.
104	LMPAE221.INT	80	
105	LMPBE221.INT	80	
			Output Files for Example Calculations
			Preparation Program Output Files for the LMPAE433, LMPAE221, and the LMPBE221 Mechanisms
106	LMPAE433.PRO	132	Preparation program listing files
107	LMPAE221.PRO	132	
108	LMPBE221.PRO	132	
109	LMPAE433.MOD	80	Mechanism data files
110	LMPAE221.MOD	80	
111	LMPBE221.MOD	80	
112	LMPAE433.FTN	80	Mechanism-dependent subroutines
113	LMPAE221.FTN	80	
114	LMPBE221.FTN	80	
			PREPEMIT Output Files for the three Mechanisms, Obtained Using CALTOT.EMI as the Representative Emissions Profile
115	CALAE433.RXP	80	Emissions parameter files
116	CALAE221.RXP	80	
117	CALBE221.RXP	80	
118	CALAE433.SAR	80	Lumped SAROAD assignment file
119	CALAE221.SAR	80	
120	CALBE221.SAR	80	
			LUMPEMIT Output File containing processed emissions for CALTOT and SOUCOA
121	CALAE433.EMO	80	Emissions for CALTOT, processed for use with the LMPAE443, LMPAE221, and the LMPBE221 mechanisms
122	CALAE221.EMO	80	
123	CALBE221.EMO	80	

Table A-1 (continued) - 7

No.	File Name	LRECL	Description
123	SOUAE433.EMO	80	Emissions for SOUCOA, processed for use with the LMPAE443, LMPAE221, and the LMPBE221 mechanisms
125	SOUAE221.EMO	80	
126	SOUBE221.EMO	80	
			Test Calculation Integration Output Files, and Calculated Concentration Data
127	LMPAE433.OUT	132	Integration listing output file using the LMPAE433, LMPAE221, and LMPBE221 mechanisms and lumping, emissions assigned using CALTOT.EMI as the representative profile. One calculation each for processed emissions for CALTOT and SOUCOA is in each output file.
128	LMPAE221.OUT	132	
129	LMPBE221.OUT	132	
130	CALAE433.CDI	80	File name "mixmech.CDI" indicates type of calculation, where "mech" indicates the mechanism and lumping approach, and "mix" indicates the emissions profile being simulated.
131	SOUAE433.CDI	80	
132	CALAE221.CDI	80	
133	SOUAE221.CDI	80	
134	CALBE221.CDI	80	
135	SOUBE221.CDI	80	

Barcode on previous page